Our Ref: 12612821

P/00/009
Regulation 3:2

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Applicant(s):

William Stephen Jenkin
71 Lindrum Crs
HOLT ACT 2615
Australia

Address for Service:

DAVIES COLLISON CAVE
Patent & Trade Mark Attorneys
255 Elizabeth Street
Sydney, New South Wales, Australia, 2000

Invention Title: **Interaction system**

The invention is described in the following statement:

INTERACTION SYSTEM

Background of the Invention

The present invention relates to a method and apparatus for providing secure interactions via a communications network.

Description of the Prior Art

The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that the prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

A number of problems exist in utilising communications systems such as the Internet. In particular, when individuals are using the Internet, or other similar networks for performing interactions, such as sending e-mail, performing financial transaction, or the like, it is possible for a user's interaction to be compromised in a number of different ways.

For example, hackers can intercept information relating to the interaction and then utilise this in a malicious manner. Thus, when a user is performing a banking transaction via the Internet, it is possible that a third party may be able to covertly obtain the individuals banking details then use these to perform fraudulent transactions.

To attempt to address these problems, it is possible to use a number of different security protocols, for example by using encrypted communications, such as SSL (Secure Socket Layer) connections, VPNs (Virtual Private Networks), or the like.

Whilst such processes can reduce the opportunity for third parties to intercept and interpret communications, this does not overcome all problems. For example, Internet content is also populated with malware such as spyware, viruses or the like, and the use of security such as SSL or VPN protocols, does not prevent such malware infecting a user's computer.

Accordingly, such malware it typically targeted using appropriate anti-virus software. However such software provides only limited protection. For example, such software will typically only operate to protect the computer on which it is installed. Accordingly, unless every terminal on the system includes appropriate anti-virus software, viruses and other malware are able to disseminate via the network. Furthermore, even when anti-virus software is provided, it is not always able to accurately detect viruses and other malware. This is due to the fact that the software typically operates utilising signatures that are indicative of viruses, and it may occur, particularly when new viruses occur, the signatures are out of date meaning the computer is vulnerable to the latest viruses even though appropriate software is installed. This form of attack, also known as a "zero-day" attack, means that the systems remain compromised until a defence, such as a suitable virus signature, is built.

Summary of the Present Invention

In a first broad form the present invention provides a method of performing secure interactions using a terminal, the method including in the terminal:

a) communicating with a service provider, the service provider being responsive to the communication to:
    i) validating the terminal to confirm the terminal is authorised for interaction; and,
    ii) establish a control connection with the terminal;
b) receiving control commands from the service provider; and,
c) performing interactions at least partially in accordance with the control commands.

Typically the method includes, in the terminal, transferring a sequence number and an identifier to the service provider, the service provider being responsive to the sequence number and identifier to at least partially validate the terminal.

Typically the method includes, in the terminal and at the service provider:

a) exchanging digital certificates;
b) mutually authenticating the digital certificates; and,
c) establishing the control channel in response to successful authentication.

Typically the method includes, in the terminal:

a) receiving a valuation function;
b) executing the valuation function to determine a value; and,
c) transferring the value to the service provider, the service provider being responsive to check the value is correct.

Typically the valuation function is for determining at least one of:

a) a checksum value based on an image of the terminal system; and,
b) a checksum value based on an image of an application.

Typically the method includes, in the terminal:

a) receiving rules from the service provider; and,
b) performing interaction at least partially in accordance with the rules.

Typically the rules are at least one:

a) base rules for controlling access to services; and,
b) application rules controlling the operation of applications.

Typically the method includes, in the terminal, and during interaction:

a) compare operation to the rules; and,
b) terminate operation in response to an unsuccessful comparison.

Typically the method includes, in the terminal:

a) receiving a request from an operator requesting launch of an application;
b) comparing the request to base rules; and,
c) sending a launch request to a service provider in response to a successful comparison, the service provider being responsive to at least one of:
   i) at least partially validate the application; and,
   ii) provide a launch command.

Typically the method includes, in the terminal:

a) receiving the launch command; and,
b) launching the application in response to the launch command.

Typically the method includes, in the terminal:

a) implementing independent execution environment(s); and,

b) implementing the application in the independent application environment.

Typically the method includes, in the terminal:

a) providing user authentication information to the service provider, the service provider being responsive to authenticate the user; and,

b) communicate with the service provider or another terminal to perform the interaction in response to a successful authentication.

Typically the method includes, in the terminal:

a) receiving a request from an operator requesting termination of an application;

b) sending a termination request to a service provider, the service provider being responsive to provide a termination command;

c) receiving the termination command; and,

d) at least one of:

 i) terminating the application; and,

 ii) deleting contents of memory.

Typically the method includes, in the terminal:

a) receiving a request from an operator requesting termination of the terminal;

b) sending a shut-down request to a service provider, the service provider being responsive to provide a shut-down command and a sequence number;

c) receiving the shut-down command and sequence number;

d) updating a sequence number stored in a store in accordance with the received sequence number; and,

e) at least one of:

 i) shutting down the terminal; and,

 ii) deleting contents of memory.

Typically the service provider includes at least one of:

a) a network service provider; and,

b) an applications service provider.

Typically the service provider implements one or more processing systems.

Typically the method includes, in the terminal, interacting via a network, the network being an independent logical network provided on at least one communications network.

Typically the method includes, in the terminal:

a) detecting an attempt to tamper with the terminal; and,
b) causing the terminal to be excluded from further interactions.

Typically the method includes, in the terminal, transferring a tamper indication to the service provider, the service provider being responsive to exclude the terminal from further interaction.

In a second broad form the present invention provides a method of performing secure interactions using a terminal, the method including in a processing system:

a) validating the terminal to confirm the terminal is authorised for interaction;
b) in response to a successful validation, establishing a control connection to the terminal; and,
c) providing control commands to the terminal, the terminal being responsive to perform interactions at least partially in accordance with the control commands.

Typically the method includes, in the processing system:

a) receiving an identifier;
b) comparing the identifier to at least one of:
   i) a list of excluded terminals; and,
   ii) a list of authorised terminals; and,
c) validating the terminal at least partially in accordance with the results of the comparison.

Typically the method includes, in the processing system:

a) receiving a sequence number;
b) comparing the sequence number to a known sequence number; and,
c) validating the terminal at least partially in accordance with the results of the comparison.

Typically the method includes, in the processing system and the terminal:

a) exchanging digital certificates;
b) mutually authenticating the digital certificates; and,
c) establishing the control channel in response to successful authentication.

Typically the method includes, in the processing system:

a) providing a valuation function to the terminal, the terminal being responsive to determine a value; and,
b) receiving the value;
c) comparing the value to an expected value; and,
d) validating the terminal at least partially in accordance with the results of the comparison.

Typically the valuation function is for determining at least one of:

a) a checksum value based on an image of the terminal system; and,
b) a checksum value based on an image of an application.

Typically the method includes, in the processing system, providing rules to the terminal, the terminal being responsive to the rules to perform the interaction at least partially in accordance with the rules.

Typically the rules are at least one:

a) base rules for controlling access to services; and,
b) application rules controlling the operation of applications.

Typically the method includes, in the processing system:

a) receiving a launch request; and,
b) at least one of:
    i) at least partially validating the application; and,
    ii) providing a launch command, the terminal being responsive to the launch command to launch an application.

Typically the method includes, in the processing system:

a) receiving user authentication information;

b) authenticating the user using the user authentication information; and,
c) allowing the terminal to communicate with the service provider or another terminal to perform the interaction in response to a successful authentication.

Typically the method includes, in the processing system:

a) monitoring operation of the terminal; and,
b) prevent further interaction in response to an unauthorised interaction.

Typically an unauthorised interaction includes at least one of:

a) generation of malware; and,
b) dissemination of malware.

Typically the method includes, in the processing system:

a) receiving a termination request from the terminal; and,
b) providing a termination command, the terminal being response to the termination command to at least one of:
    i) terminate the application; and,
    ii) delete contents of memory.

Typically the method includes, in the processing system:

a) receiving a shut-down request from the terminal; and,
b) providing a shut-down command and a sequence number, the terminal being responsive to:
    i) update a sequence number stored in a store in accordance with the received sequence number; and,
    ii) at least one of:
        (1) shutting down the terminal; and,
        (2) deleting contents of memory.

Typically the processing system is operated by a service provider, the service provider being at least one of a network service provider and an application service provider.

Typically the method includes, in the terminal, interacting via a network, the network being an independent logical network provided on at least one communications network.

In a third broad form the present invention provides a method of performing secure interactions in a network environment, using a terminal, the method including, interacting via a network, the network being an independent logical network provided on at least one communications network, the terminal operating at least partially under control of at least one processing system operated by a network provider.

Typically the network uses different addressing, routing and address resolution methods to other logical networks on the communications network.

Typically each terminal coupled to the network operates under control of the at least one processing system.

Typically the method includes, in at least one processing system:

a) detecting terminals involved outside predetermined activities; and,
b) excluding the terminals from the network.

Typically users are associated with terminals, and wherein the method includes, in the at least one processing system:

a) determining a user associated with an excluded terminal; and,
b) excluding the user from the network.

Typically the method is a method according to the first or second broad form of the invention.

In a fourth broad form the present invention provides apparatus for performing secure interactions using a terminal, the apparatus including a terminal for:

a) communicating with a service provider, the service provider being responsive to the communication to authorise the terminal and establish a control connection to the terminal;
b) receiving control commands from the service provider; and,
c) performing interactions at least partially in accordance with the control commands.

Typically the apparatus includes an identity device, for connecting to the terminal, the identity device including a store for storing at least one of:

a) a terminal identifier;
b) application identifiers;
c) communications certificates; and,
d) software applications.

Typically the apparatus includes one or more tamper detectors for detecting tampering with the housing.

Typically activation of the tamper detectors causes the terminal or identity device to be excluded from further interactions.

Typically the apparatus is used in the method of the first broad form of the invention.

In a fifth broad form the present invention provides apparatus for performing secure interactions using a terminal, the apparatus including a processing system for:

a) validating the terminal to confirm the terminal is authorised for interaction;
b) in response to a successful validation, establishing a control connection to the terminal; and,
c) providing control commands to the terminal, the terminal being responsive to perform interactions at least partially in accordance with the control commands.

Typically the apparatus is used in the method of the second broad form of the invention.

Brief Description of the Drawings

An example of the present invention will now be described with reference to the accompanying drawings, in which: -

Figure 1 is a flow chart of a first example of a process for providing secure interactions;

Figure 2 is a schematic diagram of an example of a system for providing secure interactions;

Figure 3 is a schematic diagram of an example of one of the servers of Figure 2;

Figure 4 is a schematic diagram of an example of one of the terminals of Figure 2;

Figure 5 is a flow chart of an example of a registration process;

Figures 6A to 6G are a flow chart of a second example of a process for providing secure interactions;

Figure 7 is a schematic diagram of a first example network configuration;

Figure 8 is a schematic diagram of a second example network configuration;

Figure 9 is a schematic diagram of an example of interconnections between terminals and secure servers;

Figure 10 is a schematic diagram of a logical view of different types of parallel networks; and,

Figure 11 is a schematic diagram of a logical structure of a terminal.

Detailed Description of the Preferred Embodiments

An example process for performing secure interactions will now be described with reference to Figure 1.

In particular, at step 100 the user registers to use the secure interaction system, with a terminal being configured for the user at step 110.

This initial terminal configuration process need only be performed once to allow a terminal to be configured for the user and to allow the user to be registered for using the system. The registration process may be performed in any one of a number of manners, but will typically involve validating the user's identity, for example by checking a proof of identity, such as a passport, drivers licence, or the like. Additionally a user identifier may be generated, for example in the form of biometric information, a password, or the like.

During this process, the user may also typically select applications to be provided on the terminal, with the applications being dependent on the nature of the interaction to be performed. Thus, for example, if the interaction is a secure financial transaction, the application used will typically be different to if the interaction is the sending of a secure e-mail or other communication.

Once the user's identity is validated, and an identifier produced, this information can be used together with the selected applications, to configure a terminal for the user. The terminal may be any form of device that allows the user to perform secure interactions, and may therefore be in the form of a suitably configured computer system, a custom processing system, or the like, as will be described in more detail below.

The user identifier and other configuration information may be provided on an identity device that is separate to the terminal, but which can be connected to a terminal to configure the terminal for the user. This allows the user to retain the identity device at all times, thereby preventing the terminal being used by third parties. Alternatively, the identity device may be provided to allow an individual to use any suitable terminal. Alternatively, multiple identity devices, held by different parties, may be required for an application.

As part of the configuration process, the terminal will typically be registered with a service provider, allowing it to be used. The registration generally involves providing details of the terminal to allow the service provider to identify and authenticate the terminal and user, as will be described in more detail below.

The configuration may also involve the provision of applications software on the terminal. This may be achieved in a number of ways, for example by pre-loading the applications software, or by providing the applications software on separate media. This could be achieved for example, by providing applications software on the identity device, or supply a CD or other portable media containing the software, as will be described in more detail below.

Once the terminal has been configured, it can be used to perform secure interactions. To achieve this, at step 120 the terminal is activated, which causes the terminal to automatically contact a service provider. In the event in which an identity device is provided, this may not occur until the identity device is connected to the terminal, so that the terminal is correctly configured for use.

The service provider may be implemented in any one of the number of forms, but typically includes a secure server that can communicate with the terminal via a communications infrastructure or the like.

At step 130, the service provider validates the terminal and establishes a connection with the terminal. This can be used to ensure the terminal is currently authorised to perform interaction, for example by ensuring that the terminal is registered for use, and that the terminal has not been excluded from use. The validation procedure typically involves internal checks between the service provider and the terminal, such as the exchange of digital certificates, or the like, to check the terminal is a genuine terminal. This typically therefore involves some form of mutual authentication process, such as that used for SSH (Secure SHell) interactions, although it will be appreciated that any suitable mutual authentication process may be used. The process may additionally, or alternatively, involve checking or otherwise authenticating the user, to ensure that the possessor of an identity device or terminal is a genuine registered user.

The connection between the service provider and the terminal may be used to allow control commands to be supplied from the service provider to the terminal, allowing some or all operations of the terminal to be at least partially controlled by the service provider. This is typically achieved by having the terminal generate a request when certain actions are to be performed, with the service provider responding to the request with control commands that cause the terminal to perform the action. This allows certain actions, such as launching of applications, to be controlled by the service provider, as will be described in more detail below.

At step 140 the terminal is then able to launch a specific application to allow a certain type of interaction to be performed. In general, a number of different applications may be provided on the terminal with each application being used to perform a different type of interaction.

Once the specific application has been launched on the terminal, it is typical for some form of validation to be performed at step 150. The validation may be performed by an application service provider to ensure not only that the application is a registered application, but also that the application is being genuinely operated by an authorised user, or the like. The application service provider may be the same entity as the service provider. This validation may also be performed in any one of a number of ways as will be described in more detail below.

Following validation at step 150, trusted interaction can occur between the terminal and the service provider at step 160. The exact nature manner in which this is achieved will depend on the nature of the interaction, as will be described in more detail below.

Following this, at step 170, it is typical for the system to undergo a secure shutdown. The secure shutdown typically involves clearing any memory, as well as optionally updating link numbers or the like, as will be described below, to thereby further enhance system security.

In the above example, by having the terminal specifically configured with applications, and by having the service provider validate both the terminal and application as well as the user, enhances the security of the system.

A number of additional security mechanisms that can be implemented, and a number of effects of the operation will become apparent from the description below.

In one example, the terminals do not contain any of the information required to operate, such as the configuration information, with this all being stored separately in the identity device. In this instance, the terminal is effectively a blank terminal, allowing the terminal to be used for any appropriate process, or by any suitable user, once an appropriate identity device is installed. This allows users to utilise any approved and registered terminal to perform interactions, as long as they have an appropriate identity device.

In general the system can operate in two different manners depending on the nature of the communication network and the configuration of the system.

For example, if communications are via a standard OSI Layer 3 network (*OSI Reference Model*), then in general, communications will be transferred via a network that will also include data traffic between terminals not forming part of the system. Accordingly, there is a greater risk of malware being provided to a terminal forming part of the system. Accordingly, this instance, the system is implemented using a hub and spoke architecture so that trusted terminals can only communicate with the service provider, thereby preventing communication directly between terminals.

In this instance, if information is to be transferred between terminals, this information must be routed via a service provider. This attempts to prevent malicious code, such as malware, viruses, trojan programs, spam, or the like, being distributed between terminals, as such code can be blocked by the service provider.

However, in an alternative example, the system is used to implement a parallel OSI layer 3 network (generally referred to hereafter as "OSI Layer 3A"). In this instance, whilst the network infrastructure hardware is identical, the network forms its own logical network, and is therefore sometimes referred to as a parallel network or universe. To achieve this, any devices connected to the parallel network utilise a separate routing or (possibly) secure encapsulation protocol tunnelled over the host network, together with associated addressing processes. Thus, typically terminals will be assigned an address using a subset of IP addresses that are only valid within the parallel network, and which are assigned and resolved using custom DHCP (Dynamic Host Configuration Protocol) and DNS (Domain Name Server).

In this instance, communications between trusted terminals and untrusted terminals cannot occur, as the routing, addressing and encapsulation protocols are incompatible. Accordingly, in this example, the system utilises the service provider to validate the terminal as being a trusted terminal. Once this has occurred, the service provider controls the terminals so as to allow direct communication, for example in a peer-to-peer (P2P) form of architecture.

Typically it is not possible for any actions to be performed without validation of both terminal and the application by the service provider. This allows the service provider to ensure that only trusted terminals are able to undergo interaction, thereby avoiding situations in which unknown or uncontrolled terminals utilise the system. It will therefore be appreciated that this helps prevent third parties compromising system security, for example by distributing malicious code.

In the event that a trusted terminal becomes compromised, or is used for malicious activities, the service provider can operate to exclude the terminal from further interaction, thereby maintaining system security. In addition to this, the terminal can be corrupted or destroyed, to prevent further user interaction with the terminal.

A further security enhancement is to configure the terminals such that they are only able to operate based on instructions from the service provider. Thus, for example, launching of an application can occur in response to commands from the service provider only. This effectively renders the terminals useless unless they are connected to the service provider, in which case all their actions are monitored and controlled.

Thus, at all times, operation of the trusted terminals is under the control of the service provider, allowing the terminal to be permanently excluded from the system in the event that malicious activity is detected. This prevents users attempting to use or modify the terminals independent of the secure system, which could in turn allow malicious activities to be initiated. In addition to this, trusted terminals are typically fitted with tamper detection mechanisms that allow tampering with the physical structure of the terminal to be prevented. Thus, for example, if any attempt is made to open the terminal housing, this is detected by sensors, which in turn causes the terminal to be deactivated, and removed from the list of trusted terminals, thereby preventing further interaction with the system. This avoids attempts by users to hack into the hardware and override the control of the terminal by the service provider. A service provider may share their list of deactivated terminals with other service providers to enhance their collective security.

If at any stage a terminal is removed from the system, for example, due to tampering with the physical housing, or malicious activity, it cannot be used again.

Accordingly, the above described process utilises a trusted party in the form of the terminal service provider, to validate the identity of a user, and provide an appropriately configured terminal and if necessary, identity device(s). Each time the terminal is to be used, the terminal and/or identity device coupled thereto, can be verified by the terminal service provider to ensure it is a genuine terminal issued by the terminal service provider, and optionally to ensure that the terminal has not been used for any previous malicious activity.

Once the terminal or identity device is verified, the user can select an application provided thereon. This causes the terminal to contact an application service provider (which may be the same entity as the terminal service provider) to arrange for verification of the application. Assuming that the application is successfully verified to ensure it is a genuine application, the application can be launched to allow interaction to occur. During this process, the user may be required to provide authentication information, so that the application will only launch once the user has sufficiently proven their identity and had their credentials accepted by the service provider.

By performing such checks, this prevents access to the system by unauthorised users, devices or applications. This helps ensure that the system as a whole remains secure, which in turn allows the user to remain confident that their interactions are performed in a secure manner.

Furthermore, by providing appropriate checks of the user, this removes anonymity of use, allowing malicious users of the system to be permanently barred from further use of the system, again helping to ensure the system remains secure. This therefore provides traceability allowing individuals to be tracked and permanently barred from the system, thereby further helping to prevent fraudulent or malicious activity.

An example architecture for implementing the above described process will now be described with reference to Figure 2.

In this example, the system includes a base station 201, coupled to a number of terminals 203 via one or more communications networks 202, 204 as shown. The communications networks 202, 204 may be any form of communications network and can therefore include for example the Internet, intranets, local area networks (LANs), wide area networks (WANs), or the like.

In this example, the base station 201 includes a processing system 210 coupled to a database 211. In use, the processing system 210 acts as a terminal server for validating and controlling the operation of the terminals 203, and this will hereinafter be referred to as a terminal server 210.

Processing systems 205 may be provided coupled to the communications networks 202, 204, to act as application servers 205 may also be provided to validate and/or control the operation of applications implemented by the terminals. The processing system 205 will therefore hereinafter be referred to as application servers. It will be appreciated by persons skilled in the art, that the terminal server 210 may also act as an application server.

In any event, the terminal servers 210 and the application servers 205 must be capable of communicating with the terminals 203 via the communications networks 202, 204. It will therefore be appreciated that the application server 205 and terminal server 210 may be any suitable form of processing system.

An example of a suitable processing system is shown in Figure 3. In this example the processing system includes a processor 300, a memory 301 and input/output (I/O) terminal device 302, such as a keyboard and display, and an external interface 303, interconnected via a bus 304. In use the processor 300 implements applications software stored in the memory 301 to allow the required server functionality to be provided. The external interface 303 is used to connect the processing system 205, 210 to either the communications networks 202, 204 or an external database 211, 216 as shown.

It will therefore be appreciated that the processing systems 205, 210 may be any suitable form of processing system, such as a network server or the like.

An example of a terminal is shown in Figure 4.

In this example, the terminal 203 includes a processor 400, a memory 401 and input/output (I/O) terminal 402 and an external interface 403 interconnected via a bus 404. In use, the processor 400 implements applications software stored in the memory 401 to allow desired functionality to be achieved. Typically the external interface 403 is utilised to allow the terminal 203 to be connected to one of the communications networks 202, 204, and/or to allow connection of a peripheral device, shown generally at 405. The peripheral device can act as the identity device outlined above, and is typically a USB key storage device, or the like.

In general, whilst the terminal 203 may be in the form of a computer system, it is more typical for the terminal 203 to be in the form of custom hardware adapted for performing only specific interactions. Thus, whilst a generalised computer system is capable of performing a wide range of functionality in accordance with executed application software, the terminal 203 is typically only capable of performing interactions specific to the above described system, using specific applications installed thereon. This can be used to prevent a user utilising the terminal 203 in a malicious manner, for example to create or distribute malware, viruses, spyware or the like throughout the system or otherwise subvert the applications.

If the terminal 203 is formed from a more generalised processing system, it is typical for the terminal to implement a separate application environment in which to execute applications required to provide interactions within the system described above. This is typically achieved using a virtual machine executed by the terminal 203, and this prevents such applications interacting with more generalised applications within the processing system, again helping to reduce the opportunity for implementing malicious code and enabling validation of the virtual machine image by the service provider.

It will be appreciated from this that the terminal 203 could be implemented as part of any process capable device, such as a computer system, laptop, palm top, personal data assistant, mobile phone, smart phone, or the like.

Additionally, the terminal 203 may be prevented from operating independently of the terminal or application servers 210, 205. Again, this helps reduce the opportunity for individuals to utilise the terminal 203 in a malicious manner, for example as to distribute malware, viruses, spyware or the like throughout the system, or attack by reverse engineering or debuggers.

As a further precaution, the terminal 203 is typically incorporated into a tamper detection housing. In this instance, the housing is fitted with sensors that can detect attempts to tamper with the housing, for example to expose the processing components. This could be achieved in any one of a number of manners, such as through the use of force sensors for detecting forces applied to the housing, or light sensitive detectors, for detecting removal of part of the housing.

In this example, if an attempt to tamper with the housing is detected, the terminal 203 will typically shut down, with an indication of the reason for the shut down being provided to the service provider, allowing the terminal to be prevented from further interaction with the system.

In this example, if the terminal 203 is a generalised processing system, the processing system is typically capable of independent operation when utilising applications that do not form part of the system. However, when the processing system is to implement an application using the system, the processing system launches the independent processing environment or virtual machine, and then undergoes the validation processes described above, with respect to this environment. In the event that validation is not successful, the system application is not launched, otherwise the application is launched within the environment in such a manner that it cannot interact with other environments hosted by the processing system.

An example of the process of registering to use the system will now be described in more detail with reference to Figure 5. In particular, the registration process involves authenticating the user's identity, and this is usually performed by a trusted third party. The trusted third party is usually the service provider, such as the entity implementing the base station 201, or alternatively may be an application service provider, or other trusted entity.

Thus, for example, if one of the services implemented by the system is secure banking, then banks providing services via the system may perform the registration process, allowing users to attend their own bank to register, rather than needing to attend an alternative location. However, this is not essential, and it will be appreciated that any trusted party could be used.

In any event, at step 500 the user presents themselves to the trusted party, requests a terminal and/or identity carrier and specifies required services. This may be achieved by presenting the user with a list of available services, and asking the user to make a selection. Thus, for example, the user may specify that they need to utilise the terminal 203 for performing banking interactions, as well as allowing secure e-mail communications.

At step 510 the user provides information to prove their identity, thereby allowing this to be verified at step 520. This usually involves providing documentary evidence, such as a passport, drivers licence, utility bills, or the like, which the third party can use to identify the user. In one example, this is similar to the process used to perform vetting of individuals for national security purposes.

It will be appreciated that any suitable level of check may be used, and that a combination of documents may be required to provide sufficient information to verify the user's identity to the required degree. In one example, the level of verification or authentication required can vary depending on the requested services, so that additional documents may be required if the selected services include banking as opposed to e-mail. This may be assessed on the basis of a points system or the like, with different documents being assigned different points, and the user needing a combination of documents having a sufficient points value to satisfy points requirements associated with the respective service.

In any event, if the user's identity is not verified then a notification of failure is provided at step 530. At this stage a number of actions may be taken. For example, if the user simply does not have the required number of points, the user may be allowed the opportunity to select an alternative service with a lower points requirement, or may be afforded the opportunity to provide further documents. If there are suspicions that the user is attempting to fraudulently register, this may require that law enforcement is contacted. Additionally and/or alternatively, the user may be placed on an exclusion list to prevent further attempts to register with the system.

Accordingly, the user verification process can be used to ensure that all users of the system are identified, so that in the event that malicious activity occurs, the relevant users can be permanently excluded from using the system. This not only removes malicious users, but also acts as a deterrent to reduce the likelihood of malicious activity occurring.

Once the user's identity is verified, it is determined at step 540 whether a site survey is required. A site survey may be required depending on the level of security associated with the user's requested services. Thus, for example, if the requested service is for use in a military environment, it may be necessary for a representative of the trusted third party system to attend the user's office location to survey the office and see if this meets government defence guidelines.

If a site survey is required, this is performed at step 550 with the site survey being reviewed at step 560 to determine if the site has met requirements. If this is not the case the process returns to step 530 with a notification of failure being provided. In this instance the notification of failure may involve issuing a report to allow the user to correct any deficiencies in their site.

If a site survey is not required, or the site survey is passed, then at step 570 a user identifier is created or otherwise determined. The nature of the user identifier will typically depend on the level of security associated with the requested services. Thus, at a low level the user identifier may be a username and password, whilst at a higher security level this may include biometric information, such as fingerprints, iris scans, or the like. Additional identifiers may also or alternatively be used as would be appreciated by persons skilled in the art.

At step 580 a terminal 203 or removable media 405 (forming an identity device) is configured to allow the required application(s) to operate. This can include, for example, populating the memory 401 with executable code utilised to implement the required application(s) for performing the requested services.

Thus, for example, for terminals acting as routers or filers (described in more detail below), software applications are typically pre-loaded, with the terminal also being provided with instructions to allow automatic software updates to be obtained as required.

Alternatively, if the terminal 203 is a standard computer system, this may involve providing executable code on the removable media 405, or application software in another form that can be loaded on to the terminal 203. Thus for example, software applications could be provided on a CD or other optical media, a USB device, or even loaded on demand via a network, or the like.

It is typical to require validation/verification of a software application that is to be installed onto a terminal from unknown media, before use. This can be achieved for example by using a number of checksum functions {MD5, CRC, RC4} that run on the executable file. The result of the checksum is then uploaded to the service provider for validation prior to installation of the software application.

Additionally and/or alternatively, the files may be encrypted with a secret key, with the public key being downloaded from the issuing service provider or a secure server, when the applications software is to be installed. In this instance, successful decryption of the file is proof of the file's validity, as it could only have been created by the entity with the corresponding private key.

The terminal 203 and/or removable media 405 will also typically be configured with any other information required to operate within the system including, for example, a terminal identifier, optional application identifiers, communications certificates such as SSH, X.509 certificates, VPN, SSL or the like. The terminal identifier may be used to uniquely identify each terminal, and can be based on any suitable identifier, such as a MAC (Media Access Control) address, or custom identifier specifically adapted for use with the system. Similarly, application identifiers may be used to uniquely identify applications.

Additionally, a sequence number or other one time identifier, such as a one time password or the like, may be provided, which will be used in a manner described in more detail below.

The terminal 203 or removable media 405 may also be provided with authentication information to allow the terminal to authenticate the user. Thus, this may involve the configuration of username and passwords, biometric information or the like.

In general the information stored on the terminal 203, or in the removable media 405 will be encrypted to prevent this being accessed by third parties.

In general the security level clearance will also influence the nature of the terminal 203 that can be issued to the user. Example security levels are set out below.

- Level 0 – General Purpose computer equipment with uncertified O/S (operating system)
    - Level 0.1 – Insecure "as supplied" general purpose O/S (Windows, Linux, Mac OS/X) – where the user has administrative control
    - Level 0.2 – VM system O/S hosted by Windows etc. User has administrative control of host O/S and VM system.
    - Level 0.3 - General Purpose computer where user does not have direct administrative control, but control still rests outside Secure Network Provider.
- Level 1 – GP computer with trusted/certified O/S
    - Level 1.1 – boot from a CD-ROM

  - Level 1.2 – boot from Network
  - Level 1.3 – boot from Compact Flash
  - Level 1.4 – VM hyper-visor plus Untrusted O/S in isolated Environment
- Level 2 – Tamper-proof boot
  Tamperproof Flash memory permanently mounted within computer.
- Level 3
  - 3.1 tamper-proof terminal, fixed configuration, EAL 3 certified/verified O/S and software
  - 3.2 Mobile Phone, PDA, tablets, …
    Machine identity stored in phone,
    Single or small number of applications
    user needs biometric or PIN to establish application
  - 3.3 Secure Client Gateway
    Embedded Firewall/Router O/S device
    stored machine identity [non changeable]
    key identities can be attached via removable memory device
    Password protected
- Level 4 – EAL 4 to EAL 7 O/S and application
  many different biometric and authentication options
- Level 5 – hardware separation – single application per CPU. Predictable and consistent network response/performance (Elimination of Covert Channels)

It should be noted that in the above example, level 0 terminals are not typically able to be used in the secure parallel network environment as control of such devices with an unsecured operating system could compromise security.

Finally at step 590 the terminal 203 is registered with the terminal server 210 and one or more application servers 205. This typically involves providing information regarding the terminal identifier, access rules, application identifiers, sequence numbers, and user authentication information, to allow validation of the terminal, the applications, and the user, as will be described in more detail below.

This process will also typically involve the creation of base (access) rules, which are used to control operation of the terminal 203 in use. In particular, the base rules may define for example, the types of application that can be used on the terminal, as well as other parameters relating to operation. Thus for example, the base rules may place limitations on the nature of the communication mechanism used for communication with the system. Thus, for example, the base rules may specify that the terminal can only be used if the terminal is connected via a specified connection mechanism, or is located in a predetermined location (such as the surveyed site), as will be described in more detail below. The base rules are typically stored in the database 211, and uploaded to the terminal in use, as will be described in more detail below.

This is performed to allow the terminal server 210 to validate the terminal, and similarly, to allow an application server 205 to validate applications on the terminal 203. Once this has been performed, trusted interaction can occur between the terminal 203 and the corresponding application server 205.

Operation of the system to perform a secure interaction will now be described in more detail with respect to Figures 6A to 6G.

In particular at step 600 the user activates the terminal 203. Upon power up, the terminal 203 will typically perform optional internal checks at step 610. The internal checks may be of any one of a number of our forms but typically involve having the terminal 203 generate a checksum based on information stored in the memory 401, such as a hard drive mirror image. The checksum is then compared to a predetermined checksum value, which may for example be derived during the registration process, described above with respect to Figure 5. This allows the terminal 203 to determine if any attempts have been made to tamper with the content of memory or the processing hardware, such as if attempts have been made to modify applications software, to thereby manipulate operation of the terminal 203.

If the check is unsuccessful at step 620 the process typically fails at step 630.

In one example, when the process fails, operation of the terminal 203 is simply terminated, for example by initiating a shut-down procedure described in more detail below with respect to steps 1030 to 1070 below. However, in some example, such as when the terminal 203 is being used in conjunction with high security level applications, even deactivation of the terminal may not be sufficient to provide necessary security. Accordingly, upon failure of the process the terminal 203 may be adapted to self-destruct, delete the entire content of memory, or the like, to prevent further usage of the terminal.

A further option is for the terminal server 210 to be notified of the unsuccessful check, for example, by having the terminal provide an indication of the terminal identifier. This allows the terminal server 210 to add the terminal identifier to an exclusion list to prevent the terminal operating with the system in future.

In the event that the check is successful at step 640, the terminal 203 provides a sequence number or other one time password and terminal identifier to the terminal server 210. The terminal identifier is used to allow the terminal server 210 to identify the terminal 203.

The sequence number is a value or other indication that is initially loaded during the registration process and then updated each time the terminal is used. As a result, this provides a first mechanism for allowing the terminal server 210 to check the validity of the terminal 203. In particular, the terminal server 210 maintains a list of current sequence numbers in the database 211, for each terminal 203, so comparing the received sequence number and the stored sequence number, allows the terminal server 210 to provide a first level of validity checking.

It will be appreciated from this, that sequence numbers are not necessarily sequential and that the only requirement is that they are unique to each terminal and are unique each time the terminal is used. It will therefore be appreciated that the sequence numbers may be any form of one time password, or equivalent.

At step 650 it is determined if the sequence number agrees with the sequence number stored in the database 211 and if not, the process fails. In the event that the sequence number is correct the process moves on to step 660 with the terminal and the terminal server 210 operating to exchange public certificates. Each of the certificates will be validated, in a mutual validation process so that the terminal 203 validates the certificate of the terminal server 210, and vice versa. This can be performed in any one of a number of ways, but will typically be achieved using standard SSH protocols, or other similar techniques. It will also be appreciated that whilst SSH is described, and suitable method of exchanging tokens, or certificates, to allow mutual validation of the terminal 203 and the terminal server 210, may be used. Well known techniques, such as the Diffe-Helman exchange, provide each party certainty of the others claimed identity.

If either of the certificates are determined not to be valid at step 670, the process fails at step 630. In this instance, as before, the failure may result in deactivation or termination of the terminal 203. Additionally or alternatively, the terminal identifier may be added to a list of excluded terminals 203. As a further option, a honeypot system can be used, in which the process continues in a controlled manner so that the user believes that they are undergoing secure interaction, whilst instead no interaction is performed. This can be used for example, to collect further information regarding the terminal 203 or the user, to determine if attempts are being made to make fraudulent or malicious use of the system.

If the certificates are deemed to be valid, the process moves on to step 680 with the terminal server 210 operating to establish a control channel connection to the terminal 203. The control channel connection may be in the form of suitable secure connection as will be appreciated by persons skilled in the art.

By providing the control channel in this manner, this ensures that the terminals 203 operate under complete control of the terminal server 210, so that the terminal can control absolutely the tasks that the terminal 203 can performed. This prevents users implementing software that is not authorised under the system, which inturn helps prevent the spread of malware and the like. It will also be appreciated from this that if the terminal is unable to connect to a terminal server 210, then operation of the terminal 203 is prevented.

At step 690 the terminal server downloads an image valuation function to the terminal 203. The terminal 203 executes the function to determine an image checksum value at step 700. It will be appreciated from this that the image valuation function may be any form of suitable function, such as a hash function, which operates to generate a checksum value based on a mirror image of the current memory status.

The image checksum value is uploaded to the terminal server 210 at step 710 and the terminal server operates to check if this is correct at step 720. Thus, because the terminal server 210 has information regarding the applications present on the terminal 203, and previous operation history of the terminal 203, the terminal server 210 can calculate the checksum value that the valuation function will return. In the event that the received checksum value does not agree with the checksum value calculated by the terminal server 210, this again indicates that tampering with the terminal has occurred and accordingly the process fails at step 730. Again, this is typically performed in a manner similar to that described above with respect to at step 630.

Otherwise the process moves on to step 740 with the terminal server transferring base rules to the terminal 203 for installation. The base rules are typically downloaded from the database 211 based on the terminal identifier.

At the end of this process the terminal is now activated and has been validated by the terminal server 210, so that the terminal server 210 can be confident that this is a genuine terminal 203 intended for use within the system. Additionally, by checking the checksum values and sequence number, this also allows the terminal server 210 to confirm that the terminal 203 has not been tampered with, cloned, or used outside the confines of the system, which could jeopardise system security.

At this stage, to allow the user to perform an interaction, the user requests launch of an application at step 750. In general, a specific application will be provided for each form of interaction. Thus for example if the user is to perform financial interaction with a banking or other financial institution, a corresponding banking application will be selected for launch. Similarly in the event that the user is to send secure email communications a secure email application is selected for launch.

At step 760 the terminal compares the launch request to the base rules loaded into the memory at step 740 as described above. Thus, the base rules are utilised to ensure that the terminal 203 is not used for unauthorised purposes or in an unauthorised manner.

The base rules may specify certain conditions regarding operation that must be satisfied in order for interaction to occur. This may include, for example, specifying the applications that may be utilised by the terminal, specifying the time or location, specifying the nature of connections with the server terminal 210, or the like, as well as any requirements for the user of the terminal.

Thus, for example, the terminal 203 may be capable of being connected to the system via either wired or wireless connections. In this instance, the base rules may indicate that certain functionality cannot be implemented if the terminal 203 is communicating with the terminal server 210 via a wireless connection.

Accordingly, at step 770 if it is determined that the rules are not satisfied the process fails at step 780. Again the failure may result in terminal shut-down or termination as described above. However, alternatively, the user may be presented with details of the reason why the process failed, allowing this to be overcome. Thus, if the failure is due to the terminal 203 communicating wirelessly, the user may provide a wired connection to the terminal server 210, or select a different application that may be used with a wireless connection, in which case the process can resume at step 790.

A further option is the provision of a sleep mode. In this instance, if no input by a user is detected after a certain time period, the terminal 203 enters a sleep mode. In this instance, the terminal server 210 will operate to recheck the status of the terminal 203 after a predetermined time period, and in the event that the terminal is still asleep, the terminal 210 will be shut down by the terminal server 210.

This prevents third parties from blocking communications between the terminal 203 and the terminal server 210, by simulating inaction, in an attempt to subvert the operation of the terminal 203 without the control of the terminal service provider.

Once it is determined that the base rules are satisfied, the terminal 203 sends an application launch request to an application server 205 at step 790. This may be achieved in a number of manners, such as by having details of the application server 205 embedded within the terminal 203, or by having the terminal 203 provide the request to the terminal server 210, which then forwards this to an appropriate application server 205.

The respective application server 205 used may also depend on the nature of the application, and additionally or alternatively on additional factors such as the location of the communication terminal 203. Thus, for example, it would be appreciated that a different one of the application servers 205 may be used depending on the user's location, and on the nature of the service to be performed.

At step 800 the terminal and terminal server 205 operate to exchange public certificates, or similar, with these being mutually validated at step 810, in a manner similar to that described above with respect to steps 660 and 670. In the event that it is determined that the certificates are not valid the process fails at step 780.

If the certificates are valid, the application server 205 establishes a secure connection with the terminal 203 and downloads an application valuation function to the terminal 203. The terminal 203 executes the application valuation function to determine an application checksum value which is in turn uploaded to the application server at step 840. This allows the application server 205 to check that no attempts have been made to modify the application installed on the terminal 203, in a manner similar to that described above with respect to steps 710 and 720. Thus, the application server 205 calculates an equivalent checksum value and compares the calculated checksum value to be to the received checksum value. In the event that these do not agree the process fails at step 860, in a manner similar to the failure process described above with respect to step 630.

In the event that the checksum value is correct an application launch command is transferred to the terminal 203, together with optional application rules at step 870.

At step 880, in response to the launch request, the terminal 203 creates an independent execution environment and launches the application at steps 880 and 890.

The independent application execution environment is utilised so that each application is running in a completely independent environment. This allows the terminal 203 to implement a number of applications simultaneously, but prevents any interaction between the applications. Instead, each application is only capable of interacting with a corresponding application server 205. This in turn denies the opportunity for the user to implement unauthorised applications on the terminal 203 and have these interact with authorised applications to perform fraudulent or malicious activities.

At step 900 the terminal 203 provides user authentication information to the application server 205. Whilst this is described as being performed at step 900, this may in fact be required at any stage during the interaction process, and it will be appreciated that this is for the purpose of example only.

In any event, the authentication information typically includes information such as a username and/or password, biometric information or the like, determined during the registration procedure. This information is used by the application server 205 to confirm that the user is a genuine user of the system, to avoid situations in which a fraudulent user hijacks, or otherwise fraudulently uses a genuine terminal 203. The manner of user authentication will depend on the nature of the preferred implementation, and on factors such as the level of authentication required. Thus, for example, for low security uses, password based authentication may be sufficient, whereas for high security applications, biometric based authentication may be required. As such authentication techniques are known in the art, these will not be described in any further detail.

At step 910 it is determined if the user is authenticated and if not the process fails at step 920, and again this may involve any one of a number of processes being performed, as described above with respect to step 630.

In the event that the user is authenticated, the application and the application server 205 communicate to allow an interaction or other action to be performed at step 920. This may include performing a wide range of events, such as arranging for the user to draft an email, provide an email address and forward this to the application server 205 to allow the email to be transmitted to an intended destination. Similarly, the interaction may involve the user performing financial interactions via a banking application server 205.

In any event, during and typically throughout the entire interaction process, and as shown at step 930 for simplicity, the terminal 203 monitors application operation against the application rules to ensure that the application rules are satisfied at step 940. Again, if at any stage or the application rules are contravened, the process fails at step 920.

The application rules may place restrictions on a number of different aspects of application operation, such as placing requirements on the nature of the connection between the terminal 203 and the application server 205. It will be appreciated from this that any suitable rules may be imposed depending on the nature of the application and the associated interaction being performed, as well as the preferred implementation of the system.

As long as the application rules remain satisfied, at step 950 the terminal 203 will monitor to determine if an action is occurring. If an action has not occurred for a predetermined amount of time, the terminal 203 will typically enter a quiescent state and monitor for further action.

Following this, in the event that further action is detected, the terminal 203 may require the user to undergo further authentication to ensure that the correct user is utilising the terminal, before resuming the interaction process. Alternatively this may not be required if, for example, the quiescent state has entered merely because the terminal has been unable to contact the application server 205, for example if it is out of wireless communication range.

In any event at step 970 it is determined that if the interaction is complete and if not, the process returns to step 930, to allow the interaction to continue. Thus, it will be appreciated that in general steps 920 to 970 are performed continuously until the interaction is complete. It will be also be appreciated that these steps may be performed in any suitable order.

At step 980, once the interaction is complete, the user requests termination of the application. At this stage the terminal 203 sends a termination request to the application server 205, at step 990, causing the application server 205 to transfer a termination command to the terminal 203 at step 1000. At step 1010 the terminal 203 terminates the application and deletes any associated memory content. At step 1020 the terminal determines if another application is to be launched. In general if another application is to be launched the process returns to step 750. Otherwise the user typically requests terminal shut-down at step 1030.

Once terminal shut-down is requested, the terminal 203 sends a termination request to the terminal server 2010 at step 1040, allowing the terminal server 210 to transfer a termination command and a sequence number to the terminal 203 at step 1050. The sequence number is utilised at step 640 and 650 as described above, the next time the terminal is launched, and it will be appreciated that this therefore provides a per use unique identifier for use in validating a terminal.

In particular, if a third party obtained a genuine terminal and copied the entire memory content, to thereby create a duplicate terminal, when either the genuine or duplicate terminal is next used, this will result in the creation of a new sequence number. However, only the used one of the genuine and duplicate terminals will be updated with the correct sequence number, so when the unused terminal is next used, the sequence number will be incorrect. this indicates to the terminal server 210 that there is an issue with the respective terminal, which can then be excluded from the system by adding the terminal identifier to the exclusion list. In this instance, as the duplicate terminal will inherently have to duplicate the terminal identifier of the genuine terminal, this will result in exclusion of both the genuine and duplicate terminals, therefore ensuring system security is maintained.

At step 1060, the terminal 203 stores the sequence number and deletes any memory associated with its most recent operation before shutting down at step 1070.

It will therefor be appreciated by persons skilled in the art that the above described process provides for secure interaction between a terminal 203 and one or more application servers 205. In order to ensure trusted interaction can be maintained a number of security measures are implemented within the above described process.

For example, it is typical for the terminal 203, and each application on the terminal to undergo the image checksum check to ensure that no attempts have been made to modify the terminal 203 or applications implemented thereon. The checksum check is initiated by the terminal server 210 or the application server 205 which provides a respective valuation function to the terminal 203 to allow the checksum value to be generated.

By having the terminal server 210 and the application server 205 use a unique validation function, the terminal 203 will not know what function is to be used at any one instance until the function is received. This makes it difficult for users to duplicate checksum values previously sent to the terminal or application servers 210, 205 simply because a different valuation function maybe utilised each time the terminal or an application is used.

All communication between the terminal and either the terminal server 210 or the application servers 205 is typically implemented in accordance with some form of encryption protocol such as SSH and SSL to help reduce the likelihood of transmissions being intercepted, and then interpreted or modified. Accordingly, this allows the communication to be transferred via open communications networks such as the Internet 202 without the system being comprised.

However, this is not essential. In particular, because only validated terminals and corresponding users are permitted to interact with the system, this vastly reduces the opportunity for fraudulent or malicious activity within the system. Thus, for example, even if a third party were able to intercept and determine a user's password and identifier, they would not be able to pass themselves off as the genuine user without also having access to the user's genuine terminal.

Implementation of the application, initiation of the terminal and other similar associated tasks are all performed under control of the terminal server 210 and application server 205. In other words, it is not possible to implement any of the terminal functionality unless the terminal 203 is connected to a respective one of the application servers 205 and the terminal server 210. This prevents independent operation of the terminal 203 and in turn can be used to prevent third parties tampering with terminals, thereby allowing the terminals to be used for malicious purposes.

In the event that the process fails the terminal 203 is typically excluded from the system by adding an indication of the terminal identifier to the exclusion list. This will prevent any further interaction between the terminal 203 and the remainder of the system so that in the event that any terminal has been tampered with it will be permanently banned from the system.

Further Features

It will be appreciated that the level of security provided by the system can be used to offer indemnities to users for losses incurred during use of the Service. Thus, for example, banks and other financial institutions using the system may offer a complete refund of any lost monies, when the user has correctly followed all Service Provider instructions and requirements.

It will be appreciated that the level of indemnity offered may vary depending on the level of security the user obtained for their device and its operation.

To further enhance security, organisations may arrange for the identity device to be split between two physically separate devices, each of which is held by a different individual, and associated with a respective password, or the like. In this instance, when an interaction is to be performed, each of the identity devices, and hence each of the individuals is required. This reduces the opportunity for an individual within an organisation to attempt to the maliciously use the system to the detriment of the entity.

Example Configurations

A first example configuration is shown in Figure 7. In this configuration, a single secure network provider SNP is used to provide validation for an entire network.

In this example, the secure network provider SNP is coupled to a Secure Server Gateway SS-GW, which can be used to provide onward connectivity to the Internet Net. The secure network provider is also coupled to a Firewall / Router Endpoint F/R, an Authentication & Control server AUTH, an Audit & logging server AUDIT, a Secure Server SECURE and a DNS & host resolution server DNS.

A number of secure terminals (or nodes) S-TERM are connected to the secure network provider SNP via one of the firewalls F/R. Additionally, a number of secure client gateways CLIENT GW and secure client file servers CLIENT FILER may also be connected to the firewall F/R, allowing subsequent onward connection to general purpose computers PC and computers S-PC connected to the client file server CLIENT FILER via a secure LAN.

It will be appreciated from the above description, that this allows the secure network provider to implement a parallel network, in which each of the terminals S-TERM, the client gateway CLIENT GW, and the client file server CLIENT FILER are authorised by the network to allow interaction over the network.

This therefore provides a mechanism for allowing communication between terminals over a single network, using the single network provider.

Users of the terminals PC and S-PC can interact with the file servers and gateway in the normal way, with onward connectivity to the network being via the file server CLIENT FILER and gateway CLIENT GW, under the control of the network provider. This prevents the terminals PC, S-PC being used maliciously, using the mechanisms described above. Furthermore, this arrangement allows for direct interaction between the terminals S-TERM and the file server CLIENT FILER and gateway CLIENT GW, as described in more detail above.

In the second example, shown in Figure 8, two networks similar to that shown in Figure 7 are provided, with each network having its own network service provider PROVIDER1, PROVIDER2, which are interconnected via respective firewalls as shown .

In this instance, each network provider administers a separate independent parallel network, each having its own terminals. In this instance, communication between terminals located on the same network can be direct. However, if terminals on different networks are to interact, this is achieved using a hub and spoke architecture, with all communications being transferred via the service providers PROVIDER1, PROVIDER2.

An example of the connections between service providers and terminals on different networks is shown in Figure 9.

In this example, network connections can be achieved in the form of secure tunnels from Terminals to Secure Servers over the public untrusted network. From the perspective of each terminal, the connection is a point-to-point secure link to each of the servers it can connect to. Multiple Nodes may connect to the same server and a node may connect to multiple servers.

In this arrangement, the numbering plans, addresses and host/name resolution systems used per link are independent.

An example of the logical network structures that can be implemented using the parallel networks are shown in Figure 10. In this example, the structures fall into three main styles:

- Fully Isolated.
  Each link is isolated from all others, both within the Terminal and all other Terminals. The application attached to the link can only see the Secure Server, no other devices. The Server will not route traffic to any other device. Numbering plans may be "single address" where all Terminals are given identical addresses and see the same address for the server.
  Alternatively, different link addresses per Terminal may be used, but with routing restrictions in place where only one other address , the server-end link address, is visible to the application.
  As the links are private and no traffic is routed, any address range may be used.
  Host name resolution is handled by the Secure Server. Addresses returned are any that suit the Secure Server environment.

- Local Terminal Network.
  The Terminals running an applications on a server can see other Terminals and devices within the constrained address range. The server will route traffic amongst the Terminals within the address range, but nowhere else.
  Any address range may be used, as all traffic is kept local.
  Host name resolution is handled by the Secure Server.

- Private Wide Area Network
  The Terminals running an application are given a local address by the Secure Server. The server will route traffic amongst its local Terminals and will forward traffic to other Secure Networks.
  Address ranges used must be co-ordinated between Secure Servers. As they are still separate from the public network, any addresses may be used.
  Network routes must be organised and managed between Secure Networks. Constraints

on routing may be enforced.
Host name resolution to local Terminals is handled by the Secure Server. Servers must organise the sharing of host names and address resolution.

- Public network interconnection , Full Address Space,.
  The Terminals running an application can see all public addresses, and the public address range allocation is respected.
  The Secure Server allocates an address to each Terminal and supplies Host name resolution.
  "Private" addresses may be allocated to Terminals with NAT (Network Address Translation) on the Secure Server used to map private local addresses to publicly accessible return addresses.
  The Secure Server provides network access and routing, including any firewall or traffic inspection and limiting functions.
  Host name resolution is provided by the Secure Server. It may be the full public name scheme, a proper subset or a modified mapping.

An example of the logical structure of a terminal is shown in Figure 11.

In this example, the terminal includes a number of validated applications APP*1*, APP*2*, ... APP*n*. Each of these applications runs in a controlled, isolated execution environment, and has an associated validated application identity ID*1*, ID*2*, ... ID*n*.

Operation of the applications is controlled using an authentication/audit/logging/control layer, which is connected to a Primary service provider and Auxiliary service providers via a verified link layer, to provide controlled access to both untrusted links and to control routing access

The DNS & IP-plan used for the terminal is generally unique to each validated application environment and parallel networks provided by service providers.

In general the terminals have unique permanent Identity (Serial Number, Certificates, ...) which are used in the two-way mutual authentication process with the service provider.

To achieve this, the terminal boots & verifies itself against it's primary service provider and will sleep or halt if the connection is not available. Following this, mutual verification of links occurs, with the service provider then allowing tightly constrained traffic on verified links once validation is complete. Throughout this process, shutdown will occur for any violation of designated authorised uses, which results in permanent repudiation of the compromised terminals ID's, thereby preventing further interaction.

Example Uses

Example uses will now be described.

In particular, the system can be utilised for performing any form of interaction such as messaging, email and other types of communications, as well as for performing secure transactions such as banking or other financial transactions.

Additionally, as the system can be used to configure parallel networks, this can be used to provide networks having specific types of content. This could include, for example, providing networks having content specifically designed for adults, and separate independent networks having content aimed at children. This allows for the safe distribution of adult content, without the risk of this being accessed by children, whilst providing a separate network with child directed content. The "adult" identity devices required to access adult content would require proof of age to be issued.

The system also provides a sufficient degree of security to provide for secure document transactions. In particular, this allows for documents to be transferred securely between terminals without fear of interception or modification.

In this instance, it is typical to provide a service provider or central hub that is capable of receiving documents intended for a certain location. In this instance, when a sending terminal submits a document to the hub, receipt of the document is acknowledged by the hub, with an indication of this being provided to the user at the sending terminal. Additionally a message will be transferred to a receiving terminal 203, so that when the receiving terminal is next activated, an indication of the presence of the document is provided. In this instance the document recipient can then contact the relevant hub allowing documents to be retrieved by the receiving terminal. Again once the document is retrieved an indication of this is sent to the sending terminal. For more sensitive documents, transfer may only be possible if the receiving terminal is on-line and accepting messages.

Throughout the process, the hub monitors messages transferred between the terminals, and records details of the messages, together with time stamps indicating when messages were created, transmitted and viewed. This provides an audit trail allowing the status of document transfer to be determined absolutely and effective anti-repudiation measures.

As the hub controls the terminals, this can ensure that the document cannot be accessed by terminals other than the receiving terminal, and this therefore allows documents to be securely transferred, without even requiring encryption, although typically encryption would be utilised for added security and certainty of sender identity.

In any event, this can be used to provide a level of certainty regarding document delivery that approaches that currently provided by DocEx systems.

It will be appreciated that a similar system could also be used to implement a common network for transferring Medical Records, test results and reports. In this instance, large secure datasources can operate a full secure internal LAN with a secure client File Server CLIENT FILER (as shown for example in Figure 7) accessed by PC's or terminals S-Terms. Medical records, results or reports are generated on the secure LAN PC's, uploaded to the File Server CLIENT FILER, then transmitted to a registered destination.

In this example, small offices, such as GP surgeries, have general purpose PC's connected to a secure Client Gateway CLIENT GW, which requires simple user authentication and provides file upload or download.

Connections may exist between peer secure networks, such as Hospitals, other States or Internationally, but no gateways exist into the public untrusted network.

In any event, this configuration allows files to be received by terminals S-TERM, Client Gateways CLIENT GW or other secure client File Servers CLIENT FILER, allowing the files to be viewed, with users being optionally alerted to the arrival of files.

Limited file import and export facilities exist to information providers or consumers – such as Medicare and other government organisations , off-network Hospitals and medical suppliers.

General

Accordingly, it will be appreciated that in the above described system, terminals and users are validated using an appropriate service provider, before access to network services is granted.

Validation is performed based not only on predetermined information, such as a user or device identifier, but also on information reflective of the current status of the terminal. This includes, for example, checking a mirror of the operating system or program, checking a sequence number or the like. This can be used to ensure that the terminal has not been used independently of the service provider, as well as to ensure that the terminal has not been duplicated.

Once validation is complete, operation of the terminal itself can be controlled by the service provider, using a secure control connection, allowing the operations performed by the terminal to be monitored and controlled as required.

By controlling operation of the terminals using a centralised service provider, this can be used to prevent the creation and dissemination of viruses and other malware.

Furthermore, in the event that any unauthorised activity is detected, the terminal and/or the user can be permanently excluded from the network, thereby preventing further unauthorised activities.

The term terminal is understood to encompass any processing system, such as a custom hardware device, computer system, or any other suitable processing system, such as a suitably programmed set-top box, PDA, mobile phone, or the like.

The term service provider is understood to encompass any entity, or entity operated hardware, such as a base station, server, or the like, that is at least partially involved in administering the above described process, and in particular for validating terminals and controlling their use within the process.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiments being illustrative and not restrictive.

In the context of this specification, the word “comprising” means “including principally but not necessarily solely” or “having” or “including”, and not “consisting only of”. Variations of the word "comprising", such as “comprise” and “comprises” have correspondingly varied meanings.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1) A method of performing secure interactions using a terminal, the method including in the terminal:
   a) communicating with a service provider, the service provider being responsive to the communication to:
      i) validating the terminal to confirm the terminal is authorised for interaction; and,
      ii) establish a control connection with the terminal;
   b) receiving control commands from the service provider; and,
   c) performing interactions at least partially in accordance with the control commands.
2) A method according to claim 1, wherein the method includes, in the terminal, transferring a sequence number and an identifier to the service provider, the service provider being responsive to the sequence number and identifier to at least partially validate the terminal.
3) A method according to claim 1, wherein the method includes, in the terminal and at the service provider:
   a) exchanging digital certificates;
   b) mutually authenticating the digital certificates; and,
   c) establishing the control channel in response to successful authentication.
4) A method according to claim 1, wherein the method includes, in the terminal:
   a) receiving a valuation function;
   b) executing the valuation function to determine a value; and,
   c) transferring the value to the service provider, the service provider being responsive to check the value is correct.
5) A method according to claim 4, wherein the valuation function is for determining at least one of:
   a) a checksum value based on an image of the terminal system; and,
   b) a checksum value based on an image of an application.
6) A method according to claim 1, wherein the method includes, in the terminal:
   a) receiving rules from the service provider; and,
   b) performing interaction at least partially in accordance with the rules.
7) A method according to claim 6, wherein the rules are at least one:
   a) base rules for controlling access to services; and,

b) application rules controlling the operation of applications.

8) A method according to claim 6, wherein the method includes, in the terminal, and during interaction:

a) compare operation to the rules; and,

b) terminate operation in response to an unsuccessful comparison.

9) A method according to claim 1, wherein the method includes, in the terminal:

a) receiving a request from an operator requesting launch of an application;

b) comparing the request to base rules; and,

c) sending a launch request to a service provider in response to a successful comparison, the service provider being responsive to at least one of:

i) at least partially validate the application; and,

ii) provide a launch command.

10) A method according to claim 9, wherein the method includes, in the terminal:

a) receiving the launch command; and,

b) launching the application in response to the launch command.

11) A method according to claim 1, wherein the method includes, in the terminal:

a) implementing an independent execution environment; and,

b) implementing the application in the independent application environment.

12) A method according to claim 1, wherein the method includes, in the terminal:

a) providing user authentication information to the service provider, the service provider being responsive to authenticate the user; and,

b) communicate with the service provider or another terminal to perform the interaction in response to a successful authentication.

13) A method according to claim 1, wherein the method includes, in the terminal:

a) receiving a request from an operator requesting termination of an application;

b) sending a termination request to a service provider, the service provider being responsive to provide a termination command;

c) receiving the termination command; and,

d) at least one of:

i) terminating the application; and,

ii) deleting contents of memory.

14) A method according to claim 1, wherein the method includes, in the terminal:

a) receiving a request from an operator requesting termination of the terminal;

b) sending a shut-down request to a service provider, the service provider being responsive to provide a shut-down command and a sequence number;

c) receiving the shut-down command and sequence number;

d) updating a sequence number stored in a store in accordance with the received sequence number; and,

e) at least one of:

i) shutting down the terminal; and,

ii) deleting contents of memory.

15) A method according to claim 1, wherein the service provider includes at least one of:

a) a network service provider; and,

b) an applications service provider.

16) A method according to claim 1, wherein the service provider implements one or more processing systems.

17) A method according to claim 1, wherein the method includes, in the terminal, interacting via a network, the network being an independent logical network provided on at least one communications network.

18) A method according to claim 1, wherein the method includes, in the terminal:

a) detecting an attempt to tamper with the terminal; and,

b) causing the terminal to be excluded from further interactions.

19) A method according to claim 18, wherein the method includes, in the terminal, transferring a tamper indication to the service provider, the service provider being responsive to exclude the terminal from further interaction.

20) A method of performing secure interactions using a terminal, the method including in a processing system:

a) validating the terminal to confirm the terminal is authorised for interaction;

b) in response to a successful validation, establishing a control connection to the terminal; and,

c) providing control commands to the terminal, the terminal being responsive to perform interactions at least partially in accordance with the control commands.

21) A method according to claim 20, wherein the method includes, in the processing system:

a) receiving an identifier;

b) comparing the identifier to at least one of:

i) a list of excluded terminals; and,

ii) a list of authorised terminals; and,

c) validating the terminal at least partially in accordance with the results of the comparison.

22) A method according to claim 20, wherein the method includes, in the processing system:

a) receiving a sequence number;

b) comparing the sequence number to a known sequence number; and,

c) validating the terminal at least partially in accordance with the results of the comparison.

23) A method according to claim 20, wherein the method includes, in the processing system and the terminal:

a) exchanging digital certificates;

b) mutually authenticating the digital certificates; and,

c) establishing the control channel in response to successful authentication.

24) A method according to claim 20, wherein the method includes, in the processing system:

a) providing a valuation function to the terminal, the terminal being responsive to determine a value; and,

b) receiving the value;

c) comparing the value to an expected value; and,

d) validating the terminal at least partially in accordance with the results of the comparison.

25) A method according to claim 24, wherein the valuation function is for determining at least one of:

a) a checksum value based on an image of the terminal system; and,

b) a checksum value based on an image of an application.

26) A method according to claim 1, wherein the method includes, in the processing system, providing rules to the terminal, the terminal being responsive to the rules to perform the interaction at least partially in accordance with the rules.

27) A method according to claim 26, wherein the rules are at least one:

a) base rules for controlling access to services; and,

b) application rules controlling the operation of applications.

28) A method according to claim 20, wherein the method includes, in the processing system:

a) receiving a launch request; and,

b) at least one of:

i) at least partially validating the application; and,

ii) providing a launch command, the terminal being responsive to the launch command to launch an application.

29) A method according to claim 20, wherein the method includes, in the processing system:

a) receiving user authentication information;

b) authenticating the user using the user authentication information; and,

c) allowing the terminal to communicate with the service provider or another terminal to perform the interaction in response to a successful authentication.

30) A method according to claim 20, wherein the method includes, in the processing system:

a) monitoring operation of the terminal; and,

b) prevent further interaction in response to an unauthorised interaction.

31) A method according to claim 30, wherein an unauthorised interaction includes at least one of:

a) generation of malware; and,

b) dissemination of malware.

32) A method according to claim 20, wherein the method includes, in the processing system:

a) receiving a termination request from the terminal; and,

b) providing a termination command, the terminal being response to the termination command to at least one of:

i) terminate the application; and,

ii) delete contents of memory.

33) A method according to claim 20, wherein the method includes, in the processing system:

a) receiving a shut-down request from the terminal; and,

b) providing a shut-down command and a sequence number, the terminal being responsive to:

i) update a sequence number stored in a store in accordance with the received sequence number; and,

ii) at least one of:

(1) shutting down the terminal; and,

(2) deleting contents of memory.

34) A method according to claim 20, wherein the processing system is operated by a service provider, the service provider being at least one of a network service provider and an application service provider.

35) A method according to claim 20, wherein the method includes, in the terminal, interacting via a network, the network being an independent logical network provided on at least one communications network.

36) A method of performing secure interactions in a network environment, using a terminal, the method including, interacting via a network, the network being an independent logical network provided on at least one communications network, the terminal operating at least partially under control of at least one processing system operated by a network provider.

37) A method according to claim 36, wherein the network uses different addressing and routing protocols to other logical networks on the communications network.

38) A method according to claim 36, wherein each terminal coupled to the network operates under control of the at least one processing system.

39) A method according to claim 36, wherein the method includes, in the at least one processing system:

a) detecting for terminals involved in predetermined activities; and,

b) excluding the terminals from the network.

40) A method according to claim 36, wherein users are associated with terminals, and wherein the method includes, in the at least one processing system:

a) determining a user associated with an excluded terminal; and,

b) excluding the user from the network.

41) A method according to claim 36, wherein the method is a method according to any one of the claims 1 to 35.

42) Apparatus for performing secure interactions using a terminal, the apparatus including a terminal for:

a) communicating with a service provider, the service provider being responsive to the communication to authorise the terminal and establish a control connection to the terminal;

b) receiving control commands from the service provider; and,

c) performing interactions at least partially in accordance with the control commands.

43) Apparatus according to claim 42, wherein the apparatus includes an identity device, for connecting to the terminal, the identity device including a store for storing at least one of:

a) a terminal identifier;

b) application identifiers;

c) communications certificates; and,

d) software applications.

44) Apparatus according to claim 42, wherein the apparatus includes one or more tamper detectors for detecting tampering with the housing.

45) Apparatus according to claim 42, wherein activation of the tamper detectors causes the terminal to be excluded from further interactions.

46) Apparatus according to claim 42, wherein the apparatus is used in the method of claim 1.

47) Apparatus for performing secure interactions using a terminal, the apparatus including a processing system for:

a) validating the terminal to confirm the terminal is authorised for interaction;

b) in response to a successful validation, establishing a control connection to the terminal; and,

c) providing control commands to the terminal, the terminal being responsive to perform interactions at least partially in accordance with the control commands.

48) Apparatus according to claim 47, wherein the apparatus is used in the method of claim 20.

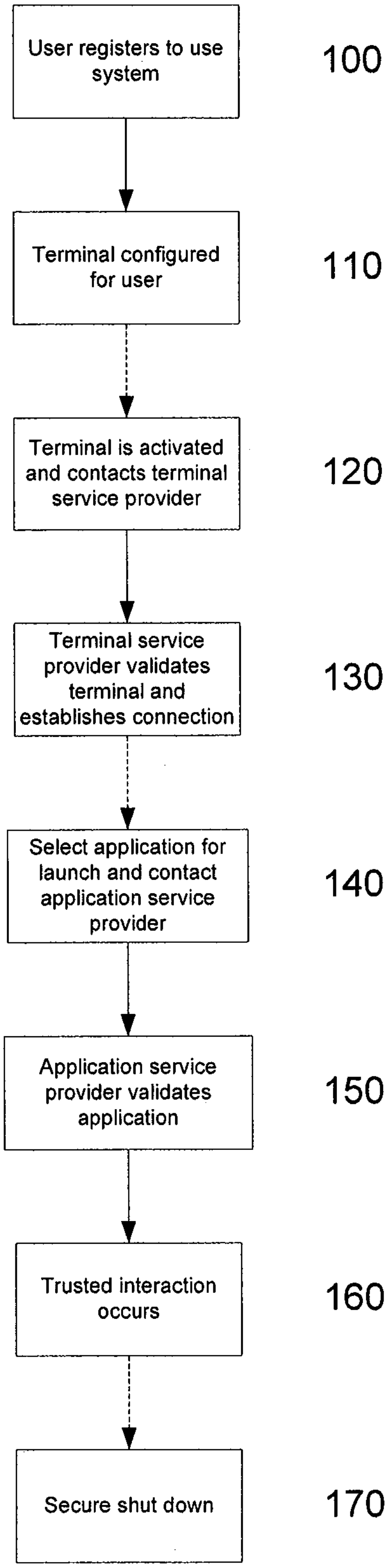
User registers to use system
100
Terminal configured for user
110
Terminal is activated and contacts terminal service provider
120
Terminal service provider validates terminal and establishes connection
130
Select application for launch and contact application service provider
140
Application service provider validates application
150
Trusted interaction occurs
160
Secure shut down
170

Fig. 1

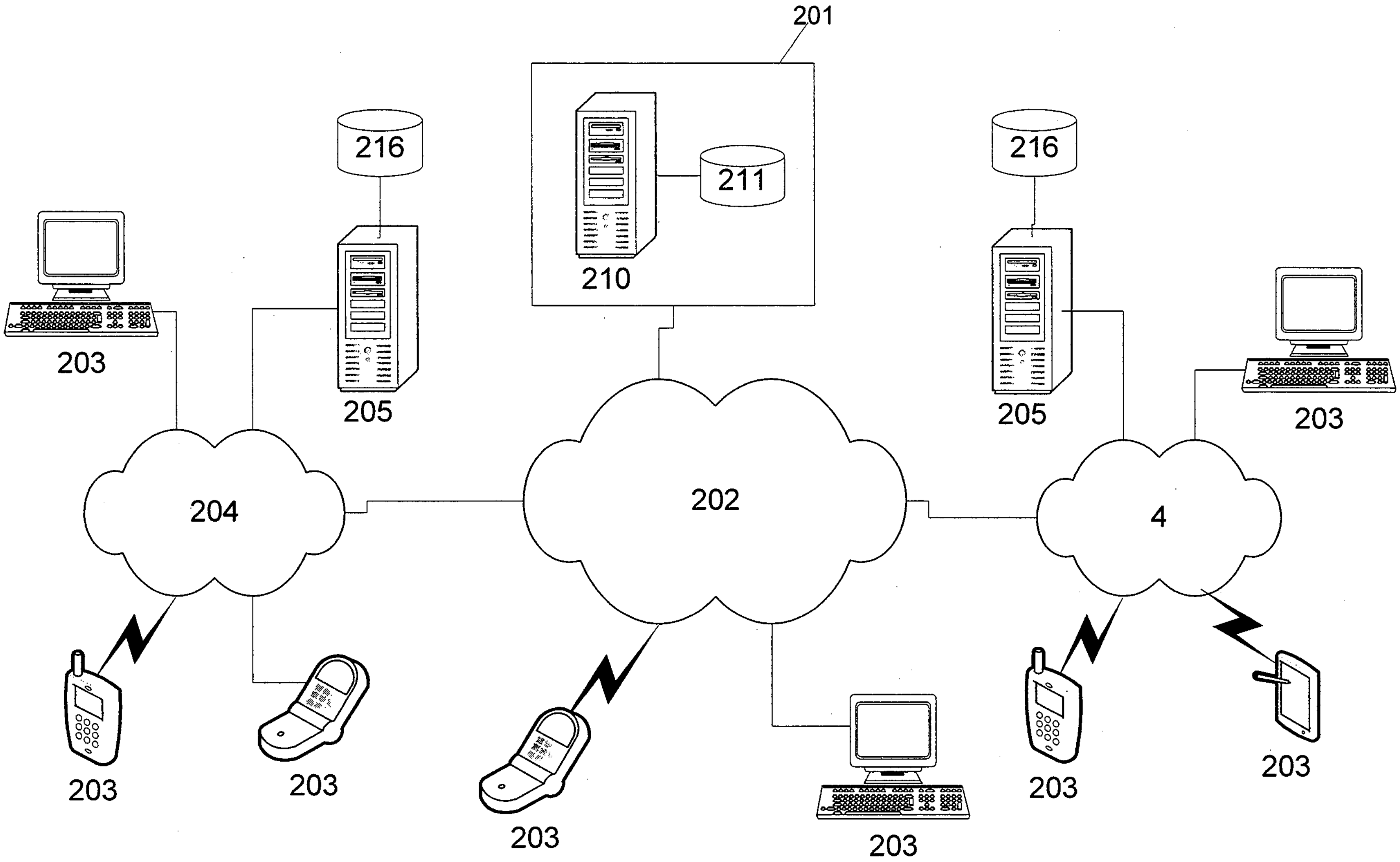
201
216
211
210
216
203
205
205
203
204
202
4
203
203
203
203
203
203

Fig. 2

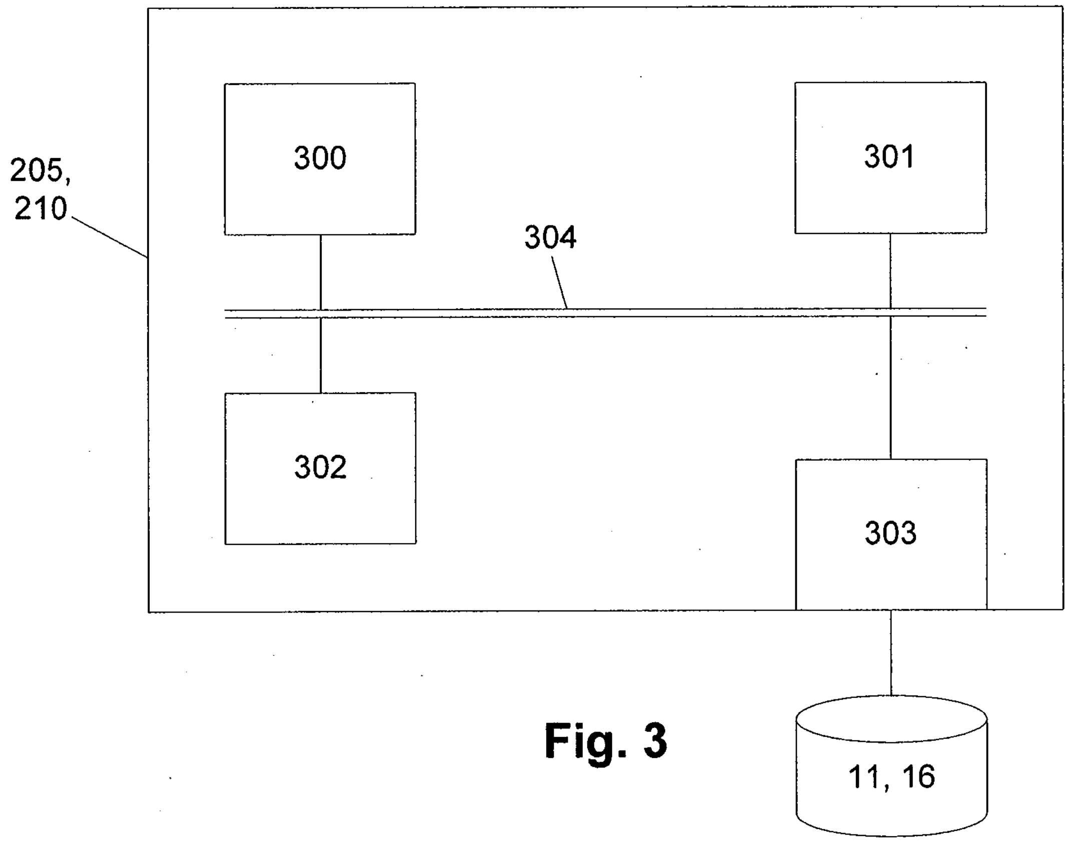

Fig. 3

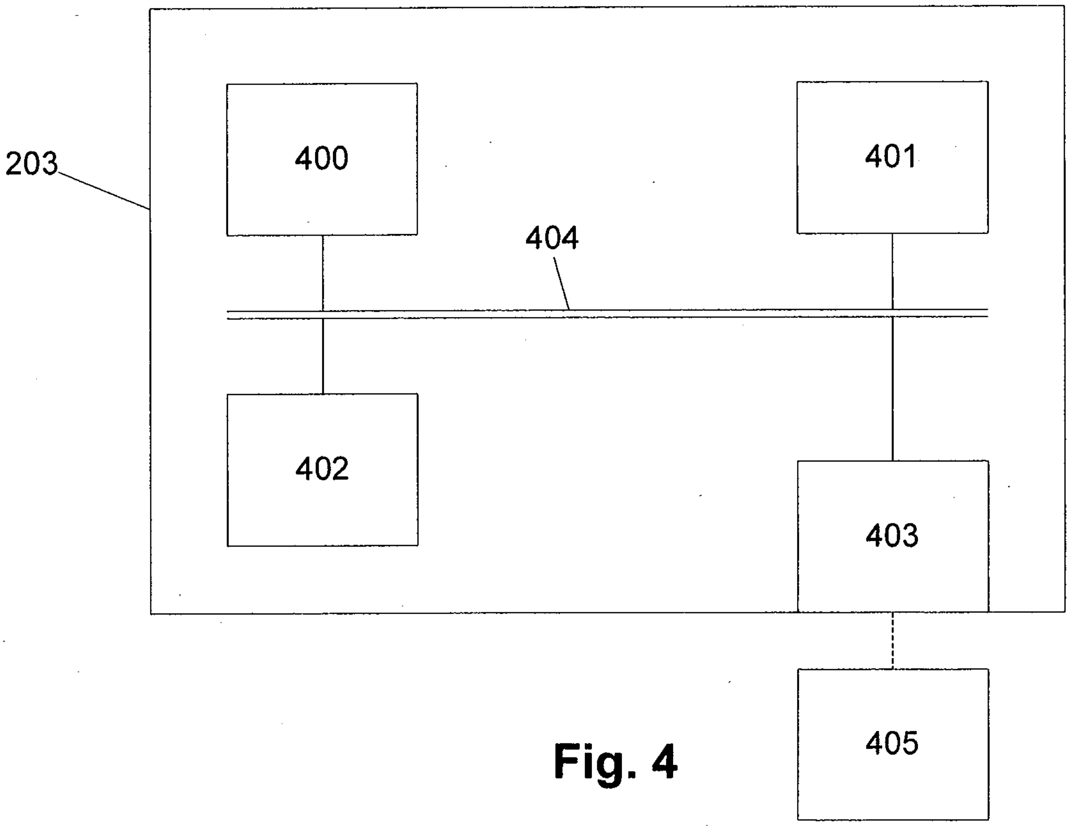

Fig. 4

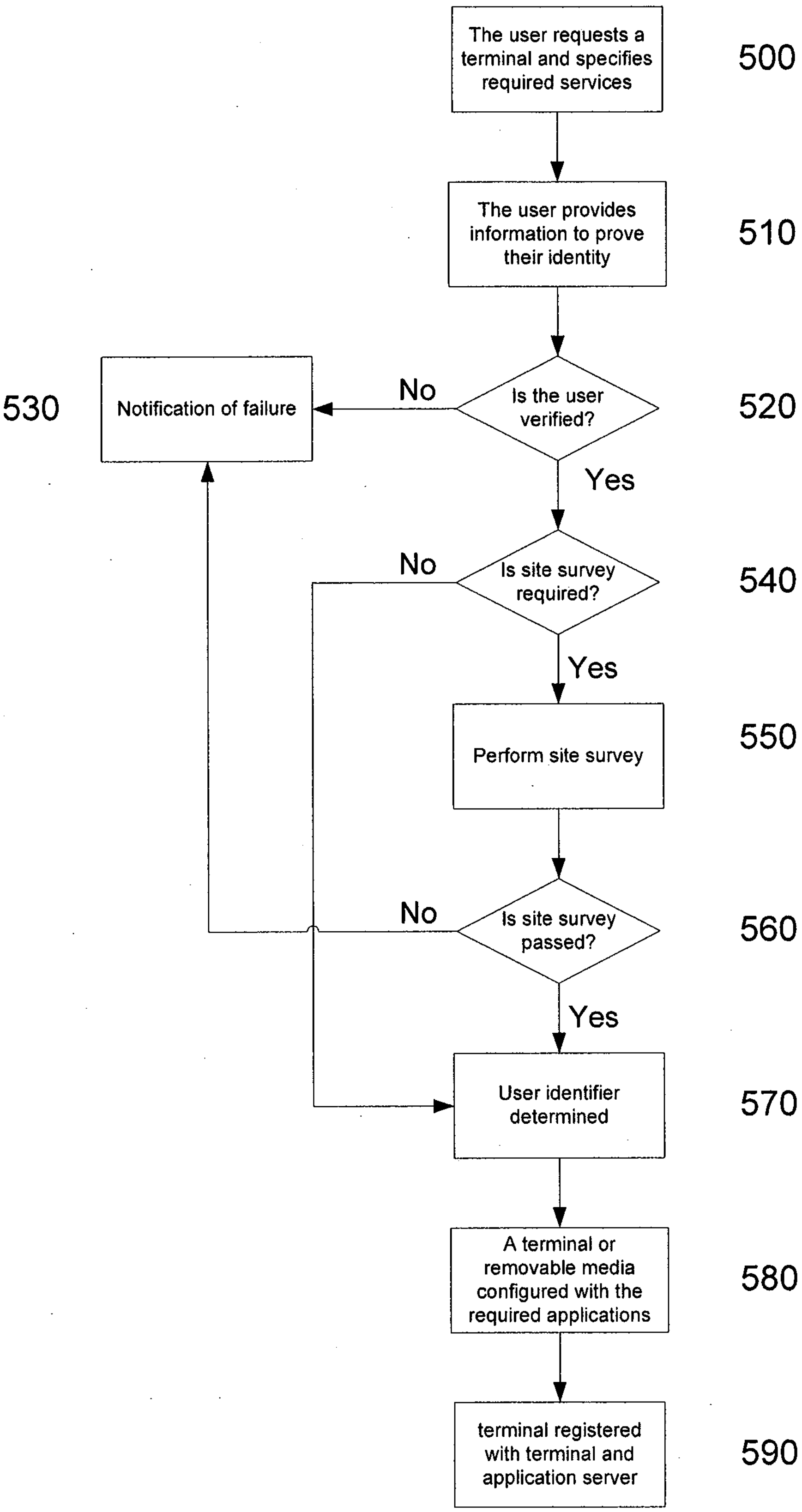
The user requests a terminal and specifies required services
500
The user provides information to prove their identity
510
Is the user verified?
520
No
Notification of failure
530
Yes
Is site survey required?
540
No
Yes
Perform site survey
550
Is site survey passed?
560
No
Yes
User identifier determined
570
A terminal or removable media configured with the required applications
580
terminal registered with terminal and application server
590

Fig. 5

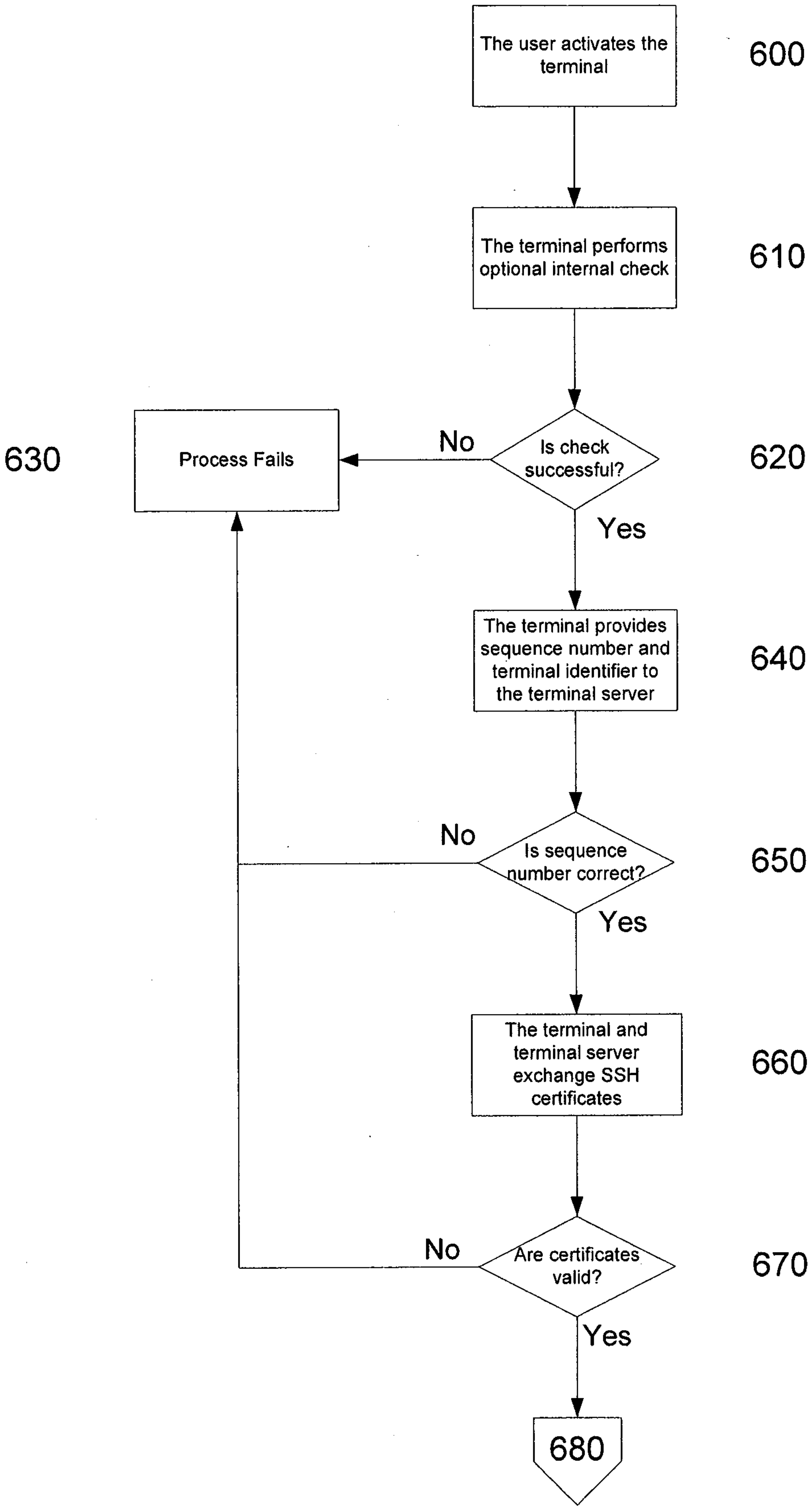
The user activates the terminal
600
The terminal performs optional internal check
610
Is check successful?
620
No
Process Fails
630
Yes
The terminal provides sequence number and terminal identifier to the terminal server
640
Is sequence number correct?
650
No
Yes
The terminal and terminal server exchange SSH certificates
660
Are certificates valid?
670
No
Yes
680

Fig. 6A

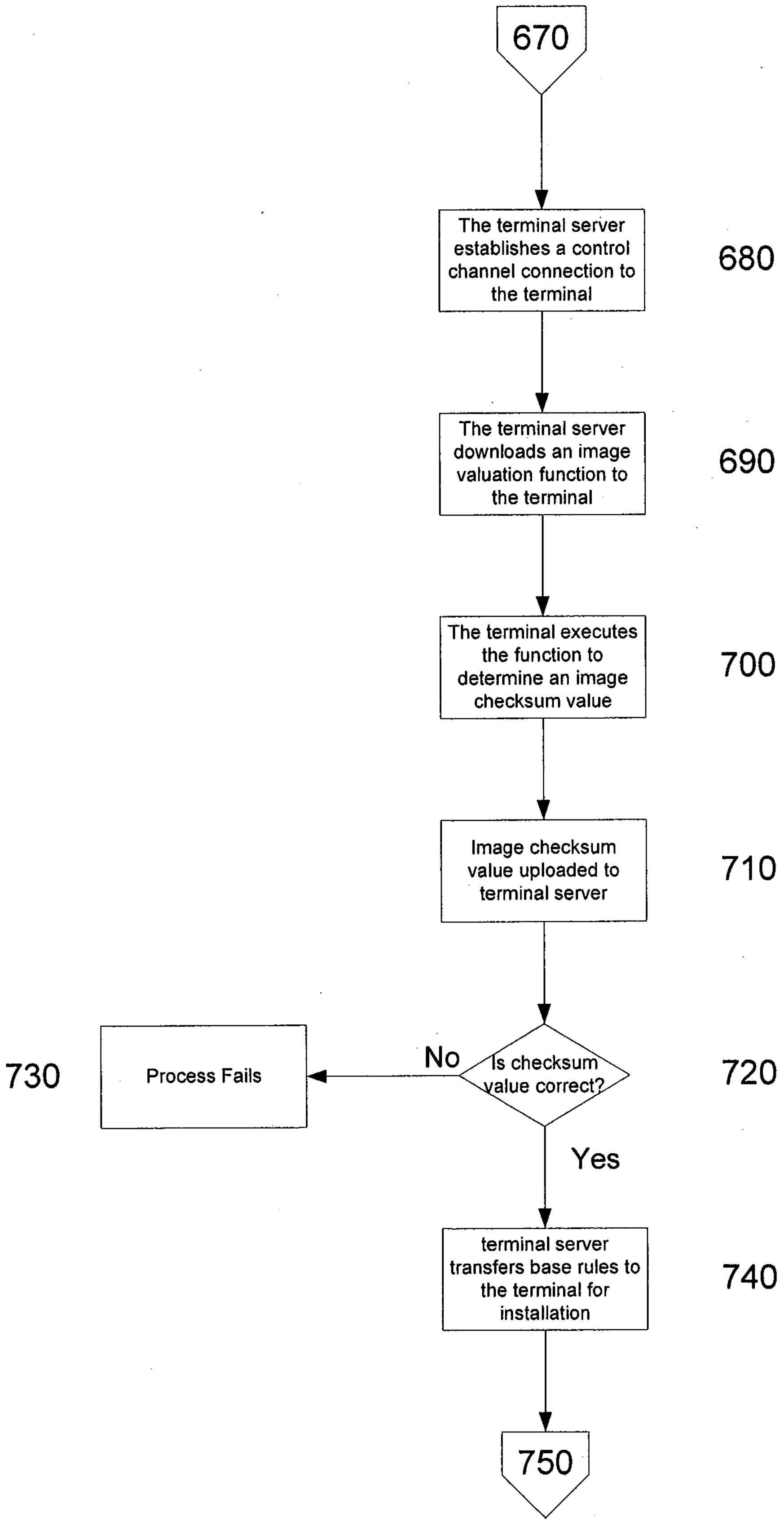
670
The terminal server establishes a control channel connection to the terminal
680
The terminal server downloads an image valuation function to the terminal
690
The terminal executes the function to determine an image checksum value
700
Image checksum value uploaded to terminal server
710
Is checksum value correct?
720
No
Process Fails
730
Yes
terminal server transfers base rules to the terminal for installation
740
750

Fig. 6B

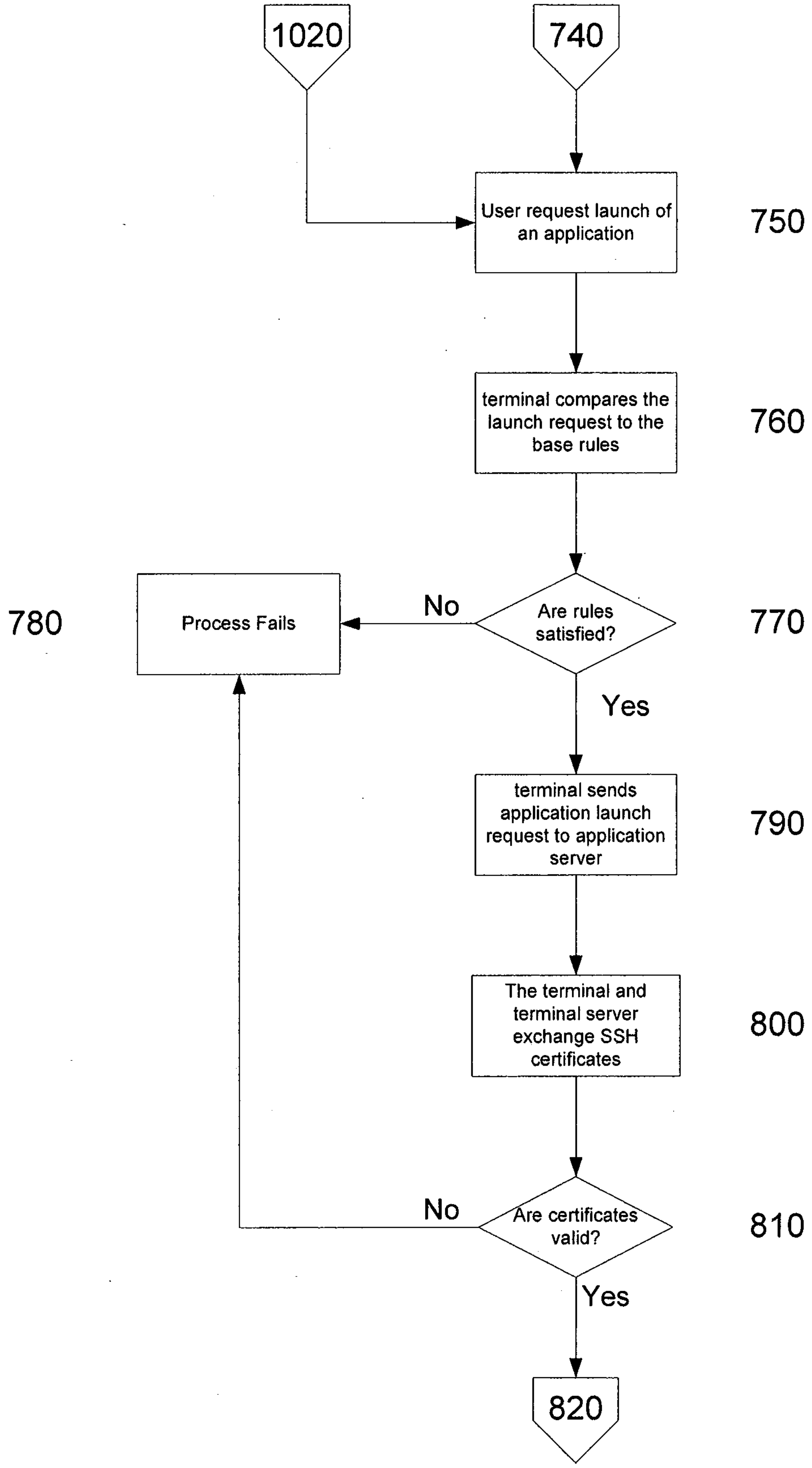
1020
740
User request launch of an application
750
terminal compares the launch request to the base rules
760
780
Process Fails
No
Are rules satisfied?
770
Yes
terminal sends application launch request to application server
790
The terminal and terminal server exchange SSH certificates
800
No
Are certificates valid?
810
Yes
820

Fig. 6C

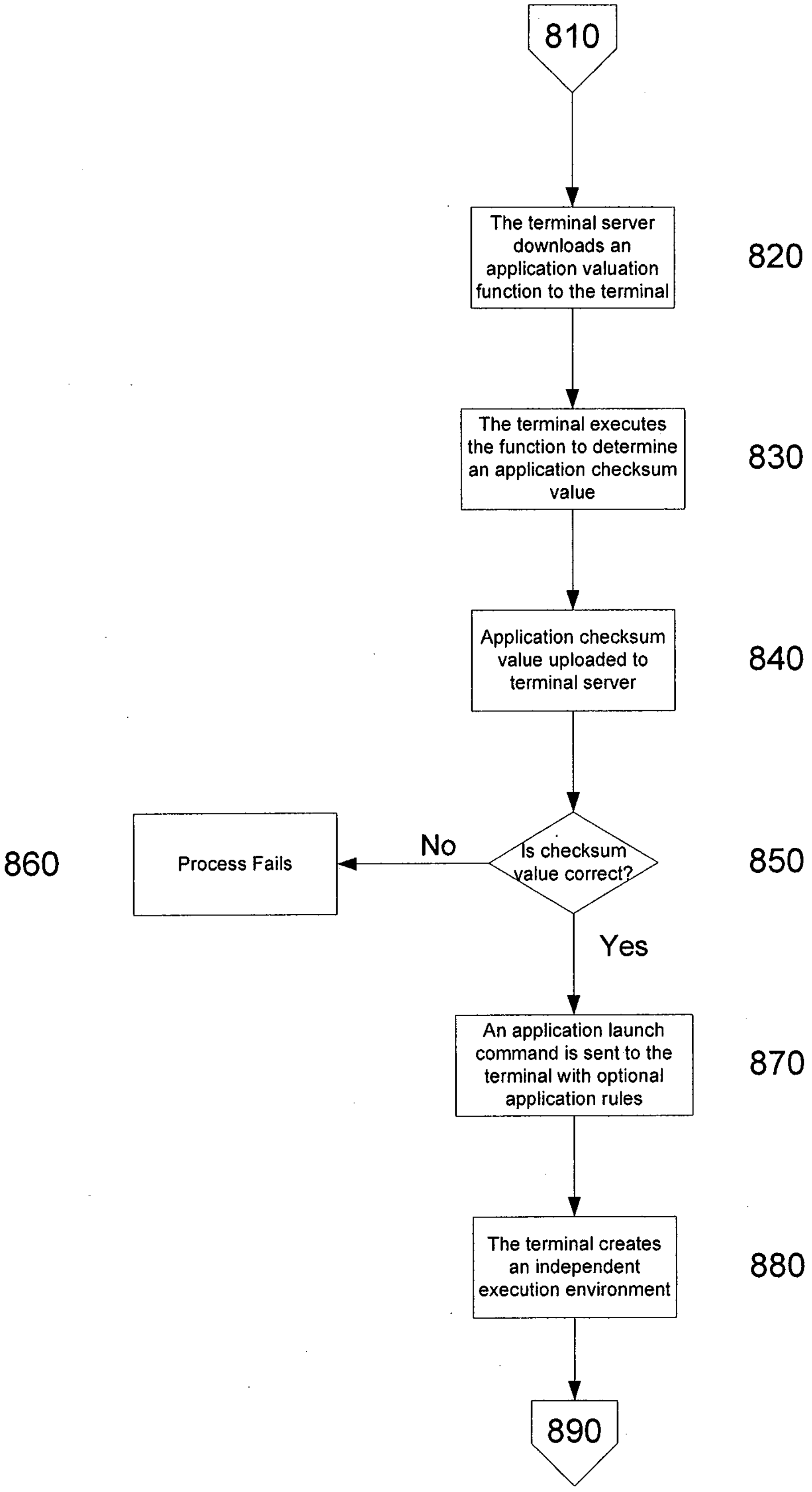
810
The terminal server downloads an application valuation function to the terminal
820
The terminal executes the function to determine an application checksum value
830
Application checksum value uploaded to terminal server
840
Is checksum value correct?
850
No
Process Fails
860
Yes
An application launch command is sent to the terminal with optional application rules
870
The terminal creates an independent execution environment
880
890

Fig. 6D

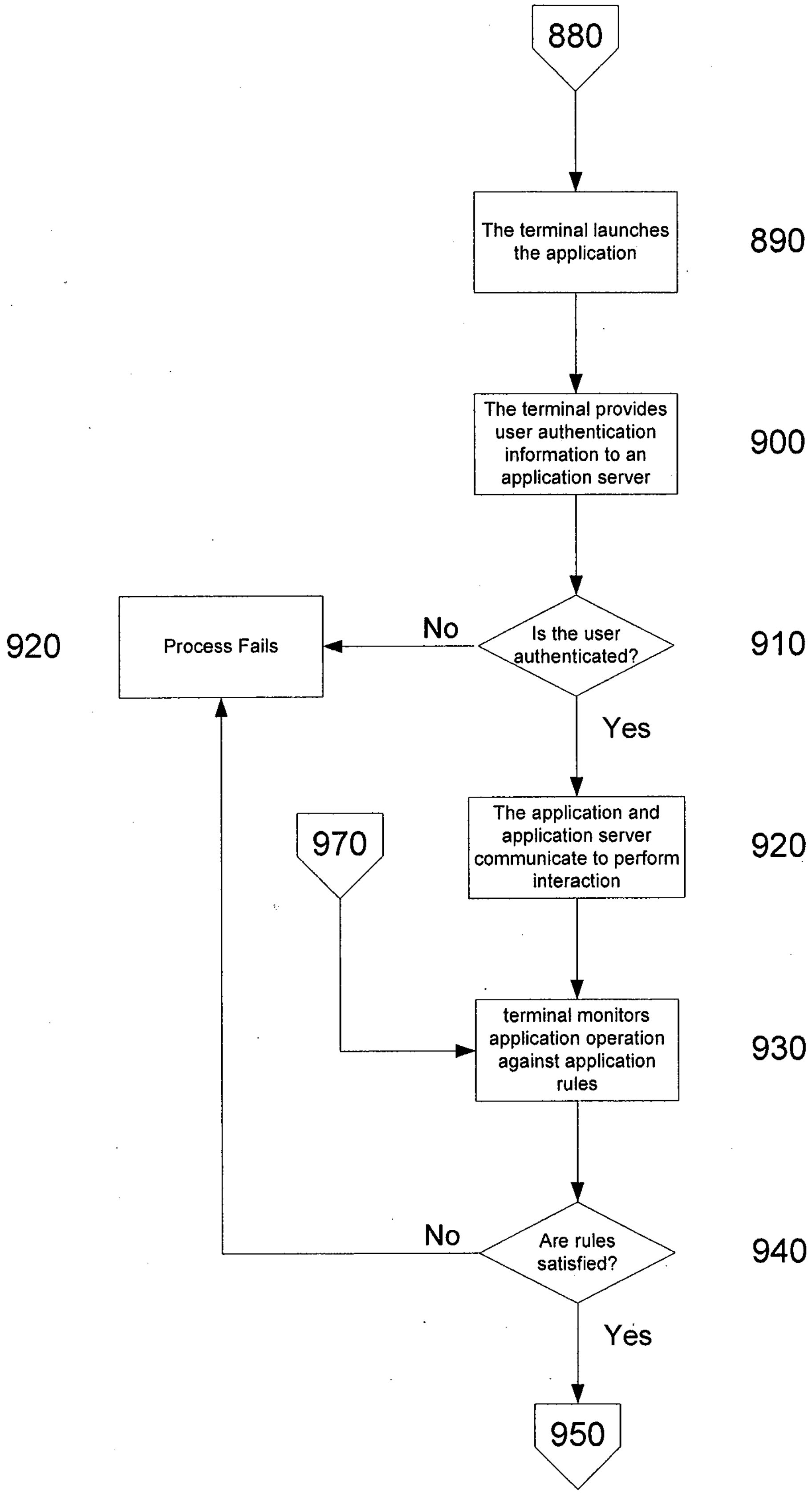
880
The terminal launches the application
890
The terminal provides user authentication information to an application server
900
920
Process Fails
No
Is the user authenticated?
910
Yes
970
The application and application server communicate to perform interaction
920
terminal monitors application operation against application rules
930
No
Are rules satisfied?
940
Yes
950

Fig. 6E

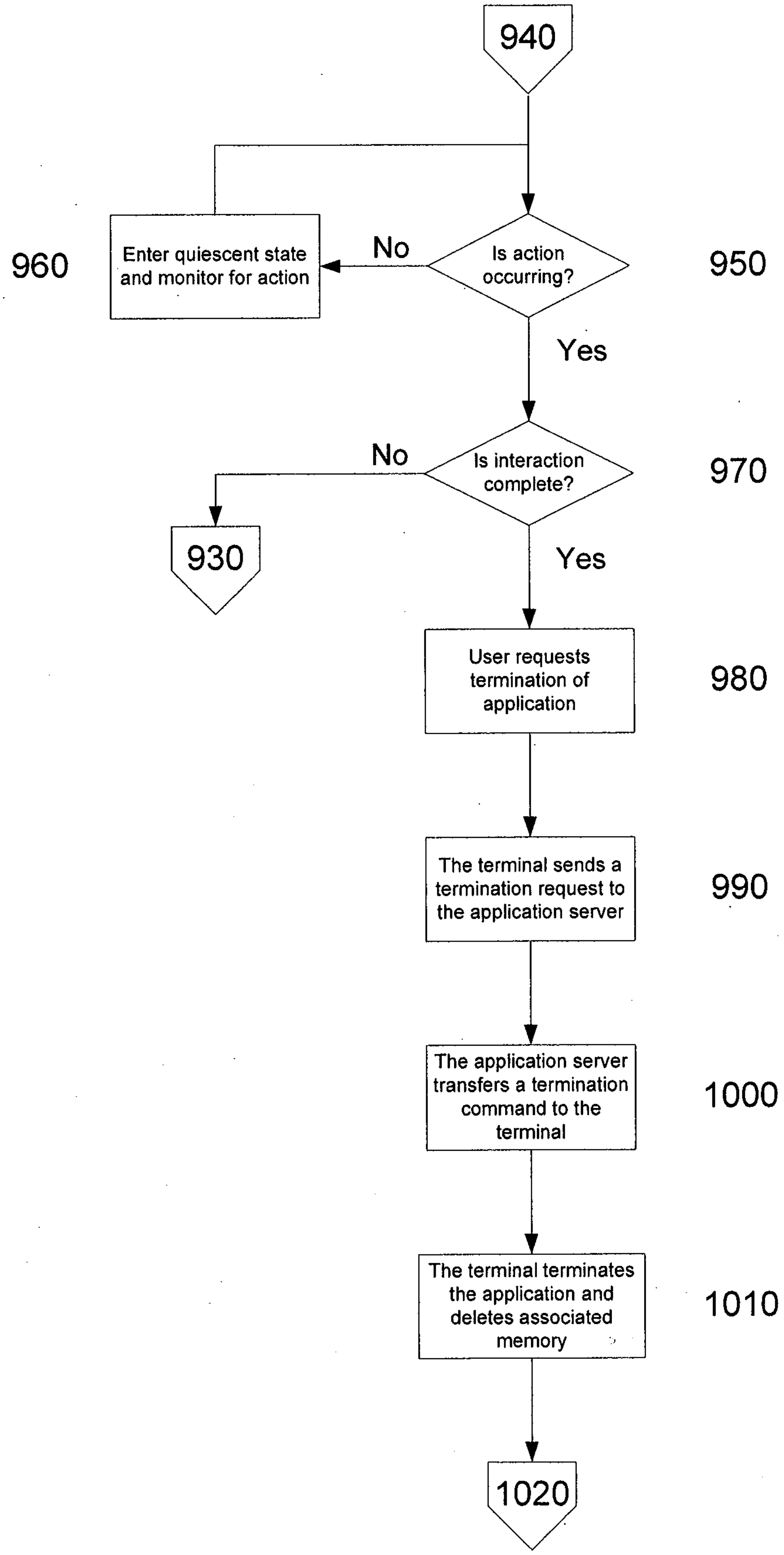
940
Is action occurring?
950
No
Enter quiescent state and monitor for action
960
Yes
Is interaction complete?
970
No
930
Yes
User requests termination of application
980
The terminal sends a termination request to the application server
990
The application server transfers a termination command to the terminal
1000
The terminal terminates the application and deletes associated memory
1010
1020

Fig. 6F

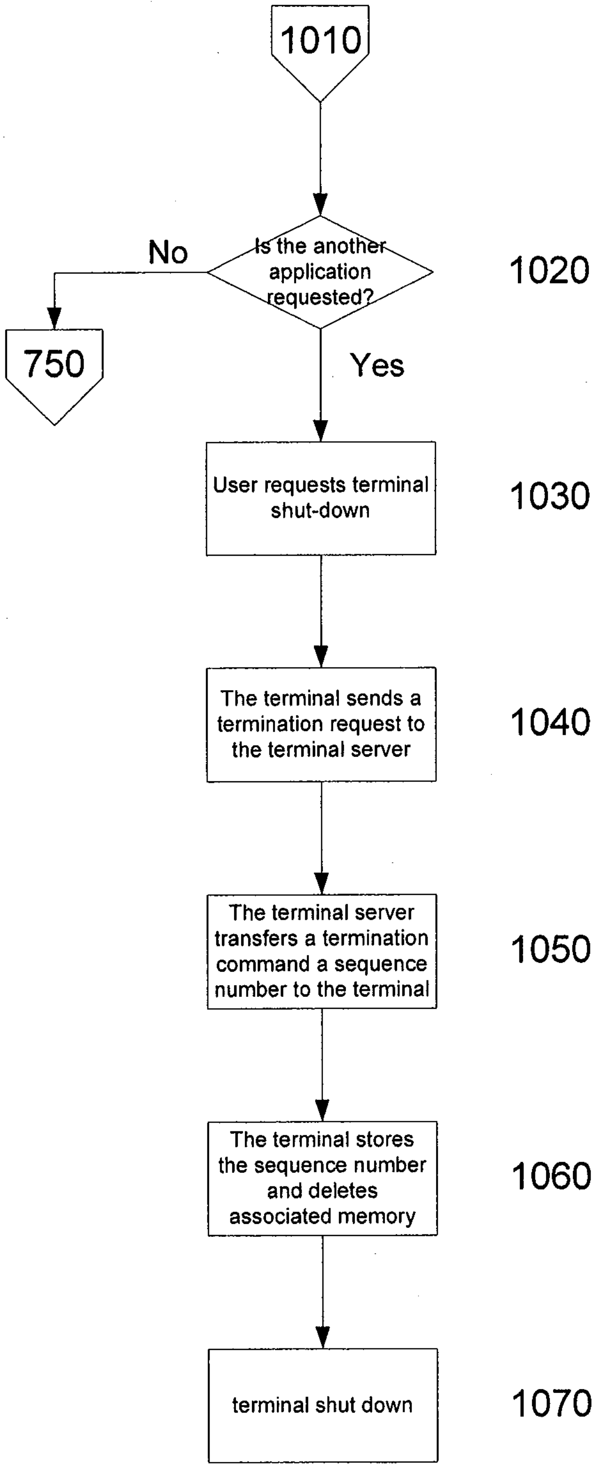
1010
Is the another application requested?
1020
No
750
Yes
User requests terminal shut-down
1030
The terminal sends a termination request to the terminal server
1040
The terminal server transfers a termination command a sequence number to the terminal
1050
The terminal stores the sequence number and deletes associated memory
1060
terminal shut down
1070

Fig. 6G

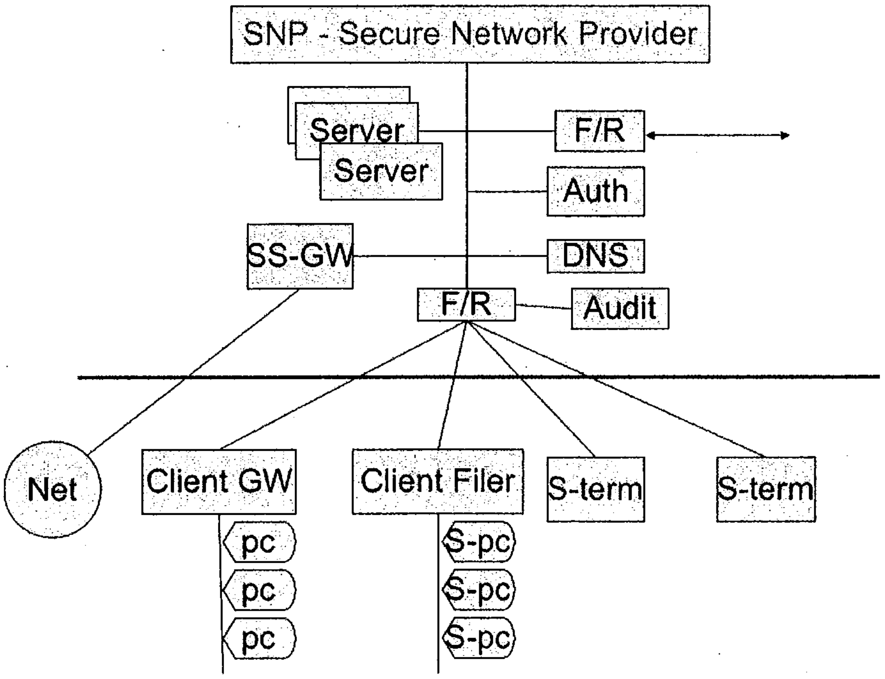
SNP - Secure Network Provider
Server
Server
F/R
Auth
SS-GW
DNS
F/R
Audit
Net
Client GW
Client Filer
S-term
S-term
pc
pc
pc
S-pc
S-pc
S-pc

Fig. 7

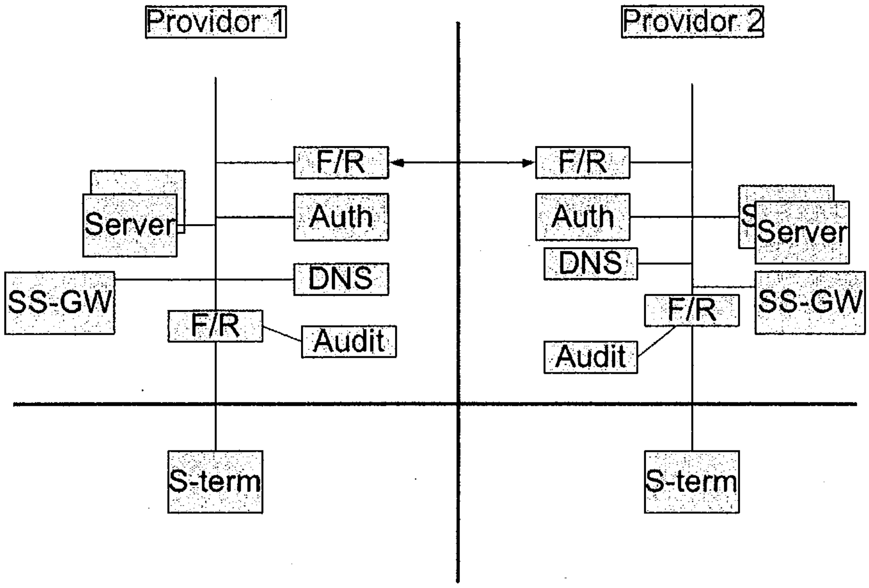
Providor 1
Providor 2
F/R
F/R
Server
Auth
Auth
Server
DNS
DNS
SS-GW
F/R
F/R
SS-GW
Audit
Audit
S-term
S-term

Fig. 8

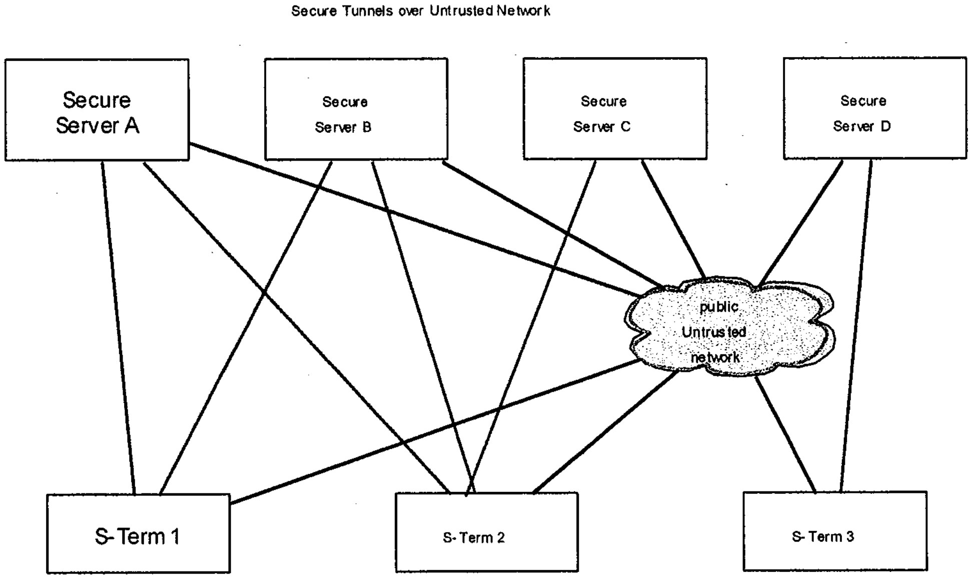
Secure Tunnels over Untrusted Network
Secure Server A
Secure Server B
Secure Server C
Secure Server D
public Untrusted network
S-Term 1
S-Term 2
S-Term 3

Fig. 9

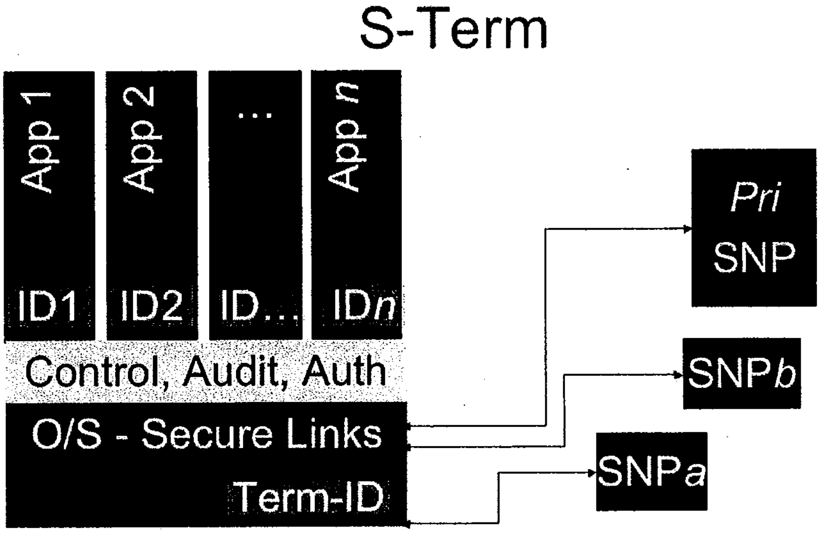
S-Term
App 1
App 2
...
App n
ID1
ID2
ID...
IDn
Control, Audit, Auth
O/S - Secure Links
Term-ID
Pri SNP
SNPb
SNPa

Fig. 11

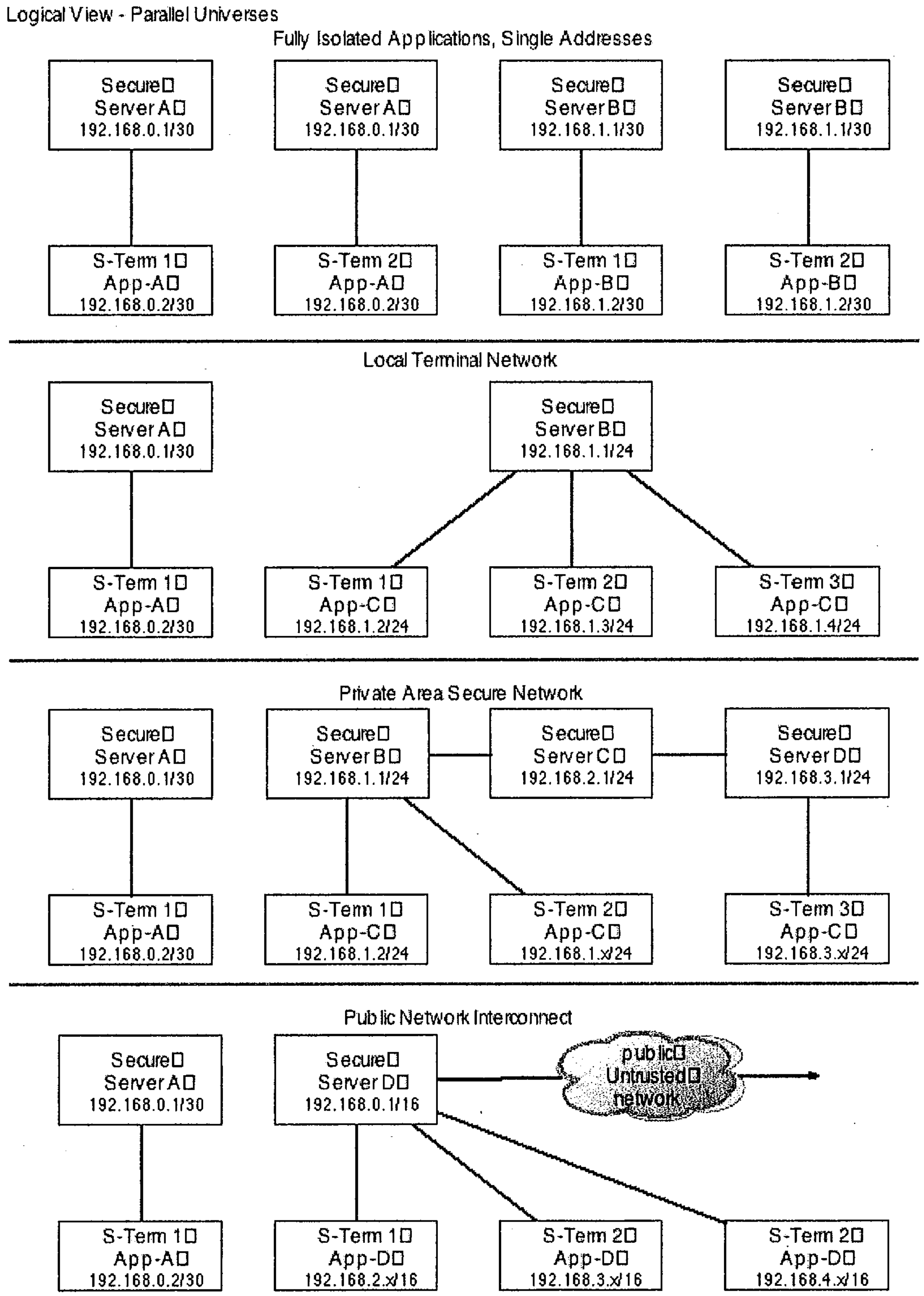
Logical View - Parallel Universes
Fully Isolated Applications, Single Addresses
Secure□ Server A□ 192.168.0.1/30
Secure□ Server A□ 192.168.0.1/30
Secure□ Server B□ 192.168.1.1/30
Secure□ Server B□ 192.168.1.1/30
S-Term 1□ App-A□ 192.168.0.2/30
S-Term 2□ App-A□ 192.168.0.2/30
S-Term 1□ App-B□ 192.168.1.2/30
S-Term 2□ App-B□ 192.168.1.2/30
Local Terminal Network
Secure□ Server A□ 192.168.0.1/30
Secure□ Server B□ 192.168.1.1/24
S-Term 1□ App-A□ 192.168.0.2/30
S-Term 1□ App-C□ 192.168.1.2/24
S-Term 2□ App-C□ 192.168.1.3/24
S-Term 3□ App-C□ 192.168.1.4/24
Private Area Secure Network
Secure□ Server A□ 192.168.0.1/30
Secure□ Server B□ 192.168.1.1/24
Secure□ Server C□ 192.168.2.1/24
Secure□ Server D□ 192.168.3.1/24
S-Term 1□ App-A□ 192.168.0.2/30
S-Term 1□ App-C□ 192.168.1.2/24
S-Term 2□ App-C□ 192.168.1.x/24
S-Term 3□ App-C□ 192.168.3.x/24
Public Network Interconnect
Secure□ Server A□ 192.168.0.1/30
Secure□ Server D□ 192.168.0.1/16
public□ Untrusted□ network
S-Term 1□ App-A□ 192.168.0.2/30
S-Term 1□ App-D□ 192.168.2.x/16
S-Term 2□ App-D□ 192.168.3.x/16
S-Term 2□ App-D□ 192.168.4.x/16

Fig. 10