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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 August 2005 (11.08.2005)

PCT

(10) International Publication Number
WO 2005/072075 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/SE2005/000107

(22) International Filing Date: 28 January 2005 (28.01.2005)

(25) Filing Language: Swedish

(26) Publication Language: English

(30) Priority Data:
0400198-8 30 January 2004 (30.01.2004) SE

(71) Applicant *(for all designated States except US)*: **MULTICOM SECURITY AB** [SE/SE]; Årstaängsvägen 1A, S-117 43 Stockholm (SE).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **GÖRAN, Eriksson** [SE/SE]; Brötvägen, Bromma (SE). **BÖRJE, Enblom** (**).

(74) Agent: **KURT LAUTMANNS PATENTBYRÅ AB**; Box 245, S-691 25 Karlskoga (SE).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: CONSTRUCTION SYSTEM FOR SUPERVISION

(57) Abstract: The invention relates to a systematic arrangement of units for the handling and transmission of alarms. Alarm events are detected in local alarm systems. The appertaining alarm information can be transmitted, via a considerable number of different communication alternatives, to central monitoring stations that have the task of monitoring and updating the local alarm systems. The systematic arrangement of units disposes over a considerable number of various means for ensuring both the safe transmission of alarm messages and the delivery of said messages to the intended recipient.

WO 2005/072075 A2

ARRANGEMENT OF UNITS TO FORM A MONITORING SYSTEM

BACKGROUND TO THE INVENTION

In today's world, there is a great need for various monitoring functions. It may be a question of monitoring the transport of valuable items (to prevent robbery) or of monitoring the transport of prisoners (to prevent escape) or of monitoring shops and premises. The latter can cover everything from detecting break-ins and fires to establishing when a line of goods runs out in an automat. From arrangements with fixed telephone lines to vehicles communicating via wireless units (e.g. mobile phones), a number of different systems can be used for monitoring. One problem that can arise is the intentional jamming, modification, recording, retransmission or other manipulation of an alarm signal by a wrongdoer. Messages may even suffer interference from natural sources or, quite simply, the "communication units" may suffer a failure. Thus, it is worthwhile establishing how well communication equipment is working and is not suffering interference or is not, for any other reason, unable to contact an alarm center or any other receiving system, e.g. a stock management system for automats.

Amongst the desirable attributes for a monitoring system are operational reliability and the ability to handle various forms of interference. The latter can be everything from natural interference (e.g. repairs to the local data network) to deliberate attempts to jam signals. Other desirable attributes for monitoring systems are: simple and cost-efficient installation; maintenance using a minimum of resources; the communication system being proof against tapping and manipulation; and, a limited communication bandwidth.

EN 50136:1998 (the industry's standard) stresses several properties for an alarm system. One important factor is transfer time, i.e. the time it takes for a signal from a local alarm system to reach an alarm center. Naturally, this should be as short as possible. It is an advantage if the system can detect the unauthorised replacement of a unit and can alarm/message such replacement so that it cannot pass unnoticed. That availability should be as high as possible is, of course, also important. Having redundant transmission paths increases the probability of

messages reaching their destinations. Other important factors in an alarm system are the abilities to monitor, check and compile statistics on: disruptions in operation; how often and when units have been inoperative; lengths of transfer times; and, other interesting details. The standard also deals with the security of information and signals in respect of, as mentioned above, the manipulation of messages and other forms of interference with signals.

DESCRIPTION

The present invention is based on an arrangement of units to form a system for the monitoring of alarms. This system comprises many different parts. Some of these embody known solutions, others embody new solutions. The combination of these parts forms a unique whole that solves many of the problems detailed above.

The system comprises: parts that enable secure communication and make use of redundant alternative paths; and, checking functions that prevent the system being affected by deliberate interference and, in a similar manner, ensure that it functions satisfactorily even when subjected to natural disruptions such as maintenance work on parts of the system.

The system has the further advantage that the software used by its parts can, in many cases, be updated in a way that does not oblige a service technician to visit the physical locations of the parts. Updating is effected by a server that communicates remotely with the units concerned. This means that several changes and new orders in a customer assignment can be administered and executed centrally. Similarly, the system can be continuously updated in a fast and efficient way when and as software improvements become available. A local alarm system can thus be externally updated via one of the communication units. This may be a question of updating the software in various parts of the local alarm system (e.g. detectors), the central unit and the communication units. This has the advantage that a lot of maintenance and updating can be effected without the need for anyone to visit the local system or for units to be sent in for checks and updates.

The system similarly possesses the advantage of secure communication. The security of communication is ensured through a number of different, advanced encryption algorithms that prevent communication being tapped or corrupted without detection. These encryption algorithms are compiled from so-called open encryption standards, but in a way that is special for systems.

Each and every unit in the system can contain a uniquely identifiable code that gives the identity of the unit. The purpose of this code is that it should prevent the unauthorised replacement of a unit with a manipulated unit. The code is encrypted and not publicly available.

The monitoring system further comprises an advanced control system to optimally distribute the traffic load by, for example, allowing certain video streams to be sent over certain channels to certain units while other destinations, with more limited bandwidth resources, can receive still images at various time intervals. It is also the case that advanced transport protocols ensure that the traffic load is limited as far as possible so that no unnecessary information clutters the communication paths.

The monitoring system comprises a number of different local alarm systems that have alarm functions and means for relaying alarm messages to one or more alarm centers. The local alarm systems can process information from various types of alarm. Alarm here means a wider term embracing everything from what is popularly referred to as an alarm (e.g. burglar alarm, fire alarm and personal attack alarm) to information on stock levels, temperature, system status, images and positioning data. When the local alarm system detects an alarm event, it has the job of sending an alarm message to an alarm center. This signal can be sent in a number of different ways using various communication methods. Examples of the latter include: the internet; fixed telephony devices that ring the alarm center when there is an alarm; and, various ways of using a wireless unit for communication. When, using one of these methods, the alarm message reaches the alarm center, an assessment is made of what action should be taken to handle the alarm message. This can be anything from sending someone to the location, to attempting contact via another communication means or alarming the police. Of

course, information and alarm signals can also be sent to other systems than an alarm center, e.g. customer systems.

Nowadays, various communication paths can be used in the management of alarm signals. An internet connection is one possible method. This is based on a unit being connected to the internet via, for example, cable, ADSL, ISDN, a mobile network, a fibre network, etc. The unit maintains two-way communication between alarm centers and the local alarm system as well as other connected systems. This communication method detects any break in the connection. However, there may be instances where an extra communication path is needed as a backup. For example, problems may arise with network overloading or repair work breaking the standard connection.

A so-called "ringing unit" is one common way of handling alarm signals. Here, when an alarm is triggered in the local alarm system, the system automatically rings the alarm center. The call may be made via the fixed telephone network or via a wireless call over a mobile network. This widely used system lacks the possibility of alarming the alarm center if the communication path is broken by, for example, the physical cutting of the fixed line or the jamming/disruption of the message to the wireless unit.

Swedish patent application SE0002743 sets out a system that includes a method for monitoring an object via a wireless connection. This system detects any breaks in the connection. To indicate its status, a wireless unit transmits messages randomly or as triggered by events. In the absence of messages from the wireless unit, the receiving server, after an automatic evaluation, sends an alarm to the alarm center. A message may have been intentionally jammed; its absence triggers an alarm. This invention requires a receiving server and a technical modification of the wireless unit. The need to modify the wireless unit is a disadvantage of this method.

Swedish patent application SE0200639 sets outs a further way of using a wireless connection to monitor an object. In SE0200639, a programmable SIM card is

inserted in a wireless unit. A configuration server can configure this card to send messages, via GPRS, to an alarm. The card can be configured by an alarm reception unit.

The monitoring system that is the subject of the present patent application can include all these communication options individually or in various combinations. Each and every one of the individual options has its weaknesses. By combining these options and integrating them into a system, many of their individual problems can be solved.

In the present invention, a further communication check has been introduced into the monitoring system. It is a device for monitoring that a wireless unit is working and able to transmit. The device comprises a server that, over a mobile network, transmits messages to the mobile units that are to be monitored. If no delivery confirmation is given for a message, the server automatically sends a connection fault alarm to a predetermined receiver or receivers. There is a free choice of communication methods for sending the alarm to the receiver. The receiver may be, for example, an alarm center, operation center or an authorised person. One advantage of this communication check is that the mobile unit does not require modification; all modifications are implemented at the central monitoring station, the server. This means that changing the number of monitored objects is quick and easy. No on-site installation is required; all technical modifications are implemented centrally in a server system. Thus, this communication check uses the "lost connection" principle of SE0002743, but with the difference that the message is sent from the server to the wireless unit whereas, in SE0200639, it is sent from the wireless unit to the server or vice versa. If the wireless unit is jammed, an alarm message is sent to one or more predetermined alarm receivers that, in their turn, decide what actions should be taken. Action may be anything from sending someone to the site, to attempting contact via another communication means or waiting a certain time to see whether contact is re-established. The mobile unit that is to be monitored can be part of a local system of objects that have to be monitored. Examples of these are: vehicles equipped with built-in mobile telephones for remote services, technical alarms, and hold-up/personal attack alarms. Even fixed installations such as alarm systems for burglary, fire, etc. and

automats equipped with mobile telephones for alarms and status reports (e.g. "restocking required") are examples of such monitoring objects. Consequently, the present device can be added to systems to increase safety/security when an internet connection is used. It can also form an addition to a fixed or mobile "ringing units" system. Here it monitors that the units are working and that messages are not being jammed.

Furthermore, the monitoring system also comprises the possibility of centrally storing images from surveillance cameras at the local alarm systems. Compared with local storage, this has several advantages. It results in fuss-free procedures at the local level. No tapes, disks or the suchlike are required at the local alarm system. In other words local installations are simple and can be managed remotely. Another very important advantage of this is that, with the images going to a so-called "trusted part", the integrity of the monitored object is protected. Neither work supervisors nor "workmates" can monitor each other. There is no risk of surveillance images going astray from taxi companies, restaurants or shops. This may result in it being easier to get permits for camera surveillance. Similarly, security is increased when images are not stored locally. Moreover, they can be monitored remotely and thus play an active role in crime prevention rather than being used as evidence after the fact.
The monitoring system also comprises units for remote control functionality. These units can remotely open doors and gates, and determine whether a visitor is authorised, etc. Thus, local systems can be controlled remotely from a central unit.

Further characterising features of the present invention are given in the patent claims below.

SHORT DESCRIPTION OF THE DRAWINGS

In the following, the present invention is described with the assistance of an example design that is illustrated in 23 figures.

Figure 1 gives an overview of a monitoring system.

Figure 2 shows a flow diagram for an alarm in figure 1.

Figure 3 gives an overview of part of a monitoring system. It comprises an alarm and monitoring function in respect of a wireless unit.

Figure 4 shows a flow diagram for an alarm in that part of the monitoring system shown in figure 3.

Figure 5 shows a flow diagram for the monitoring and configuration of a wireless unit.

Figure 6 gives a schematic overview of a part of the monitoring system.

Figure 7 is a schematic of a variant for the monitoring of a wireless unit.

Figure 8 gives an overview of how a server controls the traffic from surveillance cameras to various receivers.

Figure 9 shows a variant of the monitoring system where the local alarm system (100) is a "small system", e.g. a dwelling.

Figure 10 shows a flow diagram for alarm events, door opening and image management in such a part of the monitoring system as shown in figure 9.

Figure 11 shows a variant of the monitoring system where an alarm receiver comprises one or more mobile units.

Figures 12 and 13 show a series of screenshots illustrating the course of events on an alarm in a mobile alarm receiver.

Figure 14 shows a possible function (integrated in the monitoring system) for the remote locking and opening of an alarm terminal.

Figure 15 shows flow diagrams for three variants of how remote services, as per figure 14, can operate.

Figure 16 shows an example of a communication unit in the monitoring system.

Figure 17 gives an overview of a system solution for redundant Alive server clusters.

Figure 18 gives an overview of the functional elements in Alive server 230.

Figure 19 shows a synchronisation service in the new system solution for redundant Alive server clusters.

Figure 20 shows the synchronisation service in detail.

Figure 21 gives an overview of the functional elements in a redundant Alive server.

Figure 22 gives an overview of the functional elements in a redundant Alive server.

Figure 23 shows Online alarm management.

The 23 drawings are used in the following description of a system.

Figure 1 shows a local alarm system (100) containing various "alarm objects" (110) such as movement detectors (112.1), fire alarm (112.2), camera (113) and automats (111). A local alarm system (e.g. 100) can have various combinations of these alarm objects (e.g. 110). Alarm object 110 is in contact with a central unit (120). Communication may be through fixed lines, Bluetooth or with a wireless network, e.g. WLAN. In its turn, central unit 120 is in contact with various communication units (130). Each and every one of these communication units (130) can, individually or in combination, be integrated with the central unit. Communication between central unit 120 and the communication units (130) can be over, for example, fixed wires, WAN, LAN, WLAN, MLAN or other communication channel. The figure shows examples of such communication units (130). These can be connected up individually or together in various combinations. For example, the communication units (130) can include an internet unit (e.g. 134), a wireless unit (e.g. 131), a ringing unit (e.g. 132) and a unit (e.g. 133) connected to a fixed network. Of course, other communication units can also be used. The figure shows four communication units (130).

Internet unit 134 communicates with alarm and/or operation center 300 over TCP/IP. In the same way, it can communicate with other systems, e.g. an external alarm receiver (400) and a configuration server (250). As stated, communication here is also over TCP/IP.

Multicom unit 133 is a unit that communicates over a fixed network (which has, for example, special nodes installed in telecommunications exchanges) and can relay alarms to the alarm and/or operation center (300).

PSTN ringing unit 132 is a unit that uses the fixed telecommunications network to send alarms/information to the alarm and/or operation center (300).

PSTN ringing unit 131 is a unit that uses the fixed telecommunications network to send alarms/information to the alarm and/or operation center (300). Using wireless communication over a mobile network, wireless unit 131 can also, for example, communicate with server 230 and configuration server 210.

Server 230 monitors that wireless unit 131 in local alarm system 100 is working. Server 230 can be mirrored. Providing continuity should one of the servers go down, this increases security. Server 230 communicates with alarm receiver 400 and alarm/operation center 300 and indicates the status of the wireless units (131) it monitors. Server 230 can also communicate with configuration server 210. Configuration servers 210/250 communicate with wireless unit 131 or internet unit 134 to update and check the software in the local alarm system (100). Similarly, configuration server 210 can communicate with server 230 to set which wireless units (131) server 230 is to monitor and to which alarm receivers (400) and/or alarm/operation centers (300) it shall send the results of its monitoring. Configuration server 250 can communicate with server 240 to set how it is to regulate internet traffic from the local alarm system

Customer management devices 220/260 hold and receive information on which local alarm systems (100) are to be monitored, how monitoring is to be carried out and where communication units 130 are to send their alarm signals. In other words, which alarm/operation centers (300) are managing which local alarm systems (100) and which alarm receivers (400) are to receive messages from these local alarm systems (100). Customer management devices 220/260 set how servers 230/240 are to operate.

Server 240 regulates how traffic from internet unit 134 in the local alarm system is to be handled. For example, certain data can be sent to alarm/operation center 300 and other data to alarm receiver 400. It can also control the sending of an image from a surveillance camera (113) to any chosen destination accessible via the internet.

Monitoring device 200 is an umbrella name for units 210 – 260.

Alarm/operation centers (300) are facilities that receive alarms and information from the local alarm centers (100) and from monitoring device 200. The alarm/operation center (300) determines which actions are to be taken in the various alarm situations it may encounter.

Alarm receiver400 is a device for customers, owners, security managers, etc. to monitor a local alarm system (100).

In this arrangement, configuration servers 210 and 250 can be an individual shared unit, customer management devices 220 and 260 can be another as also servers 230 and 240.

Figure 2 shows a flow diagram for an alarm in the monitoring system shown in figure 1. The first event is the triggering of an alarm by a detector, automat or other alarm object in alarm object 110. A signal is sent from this alarm object to central unit 120. The signal can be sent via fixed networks, Bluetooth or other communication means and may be a signal or the absence of a signal. Central unit 120 processes the signals, or absences of signal, from alarm object 110. Via communication units 130, the alarm message is then sent to alarm/operation center 300 and/or alarm receiver 400. Appropriate actions to handle the alarm are instituted by these latter. Monitoring device 200 can monitor that certain of the communication units (130) are working correctly and, if they are not, send a message to alarm/operation center 300.

Figure 3 shows more closely how the monitoring system monitors wireless unit 131.

One configuration of the monitoring system is for central unit to use a PSTN ringing unit (132) in combination with a wireless unit (131). The normal path for alarms and messages is from the local alarm system (100), through the PSTN ringing unit (132) to the alarm/operation center (300). The ringing unit contacts the alarm/operation center (300) when there is an alarm. This presents no problems provided the alarm path is not broken by, for example, physical cutting of the line. This prevents the ringing unit contacting alarm/operation center 300, which remains unaware that contact has been lost.

By adding wireless unit 131 and, on top of this, monitoring device 200 (this latter comprising configuration server 210, customer management device 220 and server 230), wireless unit 131 can be monitored and alarm center 300 can receive information indicating the degree of functionality of local alarm system 300.

Via a message signal to wireless unit 131, wireless unit 131 is monitored by server 230. Wireless unit 131 returns a delivery confirmation in respect of the message signals from server 230. If the confirmation is not forthcoming, server 230 can send a report to alarm/operation center 300 and/or take other necessary action.

Theoretically, PSTN 132 can be monitored by ringing from alarm/operation center 300. However, given the constraints of reasonable practicality and cost, calls can probably only be made a few times in each 24-hour period. With monitoring of

wireless unit 131, the frequency is several times a minute. Thus, the wireless unit adds extra security.

Figure 4 shows a flow diagram for an alarm in that part of the monitoring system shown in figure 3. The first event is the triggering of an alarm by a detector, automat or other alarm object in alarm object 110. A signal is sent from this alarm object to central unit 120. The signal can be sent via fixed networks, Bluetooth or other communication means and may be a signal or the absence of a signal. Central unit 120 processes the signals, or absences of signal, from alarm object 110. Central unit 120 determines how it is to relay the alarm to alarm/operation center 300. In the described configuration, it can use PSTN 132 (a ringing unit that employs a fixed network) and/or wireless unit 131 to relay the signal to alarm/operation center 300. Central unit 200 can request confirmation that the alarm message has been received. If delivery confirmation is not given, it can try again via the unit it previously used for the alarm attempt and/or via the other unit. Server 230 remotely monitors wireless unit 131 by sending it (continuously or at intervals) message signals. If server 230 does not receive a uniquely identifiable delivery confirmation, it sends a report to alarm/operation center 300. In its turn, this determines what action to take.

Figure 5 shows a flow diagram for the monitoring and configuration of a wireless unit (131). Wireless unit 131 sends a message, e.g. an SMS message, to configuration server 210 telling the server that it wants to be installed in the system. Configuration server 210 receives the message from the wireless unit, processes the information that the wireless unit (131) sent in the message and compares it with the information that the configuration server (210) has in its database. This is to establish whether it is a valid call. If configuration server 210 judges that it is a valid call, it sends a message to wireless unit 131. This message contains the information necessary for the wireless unit to establish a communication channel with a server (230). When server 230 and wireless unit 131 have established a communication channel, server 230 monitors that wireless unit 131 is up and working. This monitoring can be effected both through server 230, as per a schedule, sending messages to wireless unit 131 (with a request for delivery confirmation) and/or wireless unit 131 sending messages, as per a

schedule, to server 230. If server 230 loses contact with wireless unit 131, server 230 sends a report to alarm/operation center 300 and/or alarm receiver 400. These, in their turn, judge what action is necessary.

Figure 6 shows a schematic of monitoring device 200. Its constituent elements can be located at various sites and can communicate with each other over various networks. The role of configuration servers 210/250 is to configure the various parts of the monitoring system. This may be a question of controlling: how often polls are sent and to whom; where alarms are to go; where images are to be presented; and, informing cameras how much bandwidth the receiver has available. The servers get their instructions from customer management devices 220/260. These manage the information detailing what the customer has ordered. This management can be effected through the devices' network connections. If they need to communicate wirelessly, they can use wireless unit 270.
Server 230 monitors wireless unit 131 in the local alarm system. Server 240 deals with the communication between internet unit 134 and alarm/operation center 300. It also functions as a switch and controls traffic from the local alarm system to the various receivers.

Figure 7 is a schematic of a variant for the monitoring of a wireless unit (131). A wireless unit (131) contains a programmable SIM card (131.1) that has a polling program (or a program that can control the unit's – telephone's – computer section) to enable remote programming by configuration server 210. There is a free choice of communication protocols (e.g. SMS) for this programming. Via the SMS card, the computer section can be programmed so that the wireless unit sends messages to server 230 and thereby implements monitoring to check that wireless unit 131 is working. Alternatively, it can be programmed to accept polls from server 230. In the absence of a poll or of a poll delivery confirmation, server 230 sends an alarm to alarm/operation center 300.

Figure 8 gives an overview of how a server (240) controls the traffic from surveillance cameras (113.x) to various receivers. These surveillance cameras (113.x) can be sited at various places and in different systems. From surveillance camera 113.1, an image or a video stream is sent over, for example, the internet.

Server 240 determines to which receiver(s) the images/video streams are to be sent. Server 240 similarly determines the quality and updating frequency with which images are to be sent to each of the receivers. Consequently, one receiver can receive a video stream while another can receive individual stills at certain intervals. The image/video stream receiver can be central image storage unit 500, alarm/operation center 300, alarm receiver 400 or any other selected receiver. Central image storage unit 500 stores digital pictures and video streams from various cameras.

Even small "systems" such as, for example, dwellings and small companies, require secure and monitored alarm transmission. Figure 9 shows a possible design of the monitoring system where the local alarm system (100) is, for example, a dwelling. The dwelling is equipped with an alarm transmitter, which may be a wireless unit (131). The alarm transmitter has the ability to transmit several different categories of alarm messages, e.g. burglary, fire, etc. The alarm transmitter is also able to transmit images, sound and data. One or several detectors (112) and one or several cameras (113) are also sited in/at the dwelling. There is direct communication with an alarm center (300) such as a police station or SOS Alarm.

The property owner/customer (400) can, via his/her own receiver (which can be a mobile telephone, palmtop computer or other unit with equivalent capabilities), receive information messages in the form of alarms and status reports on whether the connection is working.

Communication can be via, for example, a mobile network (600) such as GSM, GPRS, etc. Using encryption and firewalls, the communication is held separate from the public part of these networks. Of course, other communication methods can also be used. Alternatively, the customer/property owner (400) can also receive information via the internet and a stationary computer at a different location, e.g. work. The advantage of using the mobile network is flexibility.

Monitoring that the alarm connection is working can be effected in the same ways as previously described in the present application. Alarm transmission channels

and monitoring channels are both checked to determine the degree to which connection is active or broken.

In this way, a user always has full insight into alarm transmission status. If there is a break-in or other incident, an alarm is, of course, sent directly. If, for any reason, the alarm system loses contact with alarm center 300, the alarm center becomes aware of this on the expiry of the time period set by the user. This makes it possible to quickly take the necessary action before aggravated damage occurs.

If any door in the dwelling is equipped with a lock of the electric strike or motorised type, the owner can, thanks to the monitoring system, also open the door for others without being at home himself/herself. Transmitting images to the customer's/property owner's (400) receiver enables checking of whether the right person has been admitted. This solves, amongst other things, the problem of admitting workers when the owner cannot be at home. A further possibility would allow the owner to see various interior shots of his/her house or flat in the window of, for example, a mobile telephone.

Figure 10 shows a flow diagram for alarm events, door opening and image capture in a system as per figure 9.

Alarms

Detector 112 and camera 113 have registered an alarm. Via wireless unit 131 and mobile network 600, central unit 120 (or equivalent) sends an alarm to server 230. Server 230 relays the alarm to alarm center 300 or, via an internet TCP/IP connection and mobile network 600, to alarm receiver 400 (the customer/property owner). Alarm center 300 and alarm receiver 400 (the customer/property owner) have now received an alarm message from detector 112 and images from camera 113.

Opening doors (cf. alarms)

The bell connected to door 114 and camera 113 have both been activated. Via wireless unit 131 and mobile network 600, central unit 120 (or equivalent) sends the signal to server 230. Server 230 relays the signal to alarm receiver 400 (the

customer/property owner). Alarm receiver 400 (the customer/property owner) has now received the signal from door 114 and images from camera 113.

Via mobile network 600, alarm receiver 400 (the customer/property owner) sends an "open door" command to server 230. The server checks for authorisation and, if this is in place, the signal is relayed (via mobile network 600, wireless unit 131 and central unit 120) to door 114. The door is opened by, for example, a motorised lock.

Image capturing

Via mobile network 600, alarm receiver 400 (the customer/property owner) sends a "capture image" command to server 230. The server checks for authorisation and, if this is in place, the signal is relayed (via mobile network 600, wireless unit 131 and central unit 120) to camera 113. The images from this camera can now be viewed.

The camera can also be connected directly to wireless unit 131 or to internet unit 134 or to an ordinary, commercial internet connection.

Figure 11 shows another variant of the monitoring system where an alarm receiver 400 comprises one or more mobile units 450. This variant may be suitable for the "guard group". In other words, it may be suitable for use where, for example, the arrangement forms a "guard group". In this case, the guard group acts as a mobile alarm center where the guard is the primary alarm receiver. On the technical side, a server system receives logs and distributes alarms. The server system also monitors and logs other events, e.g. delivery confirmations, reports, reference timings, etc. In this variant, traditional alarm center 300 is the secondary alarm receiver. Via the server system, the owner (400) of the alarm system can have information sent to his/her mobile phone, computer, etc.

When there is an alarm, the server system logs and formats the alarm information for the mobile unit and (as determined by the programmed conditions in relation to assignment type, position, etc.) relays it to the "correct" guard. The server system monitors and controls the process. Alarms can be sent simultaneously to several

guards in the area. The one who acknowledges first is given the assignment (and it is cancelled for the other guards). Fire and emergency alarms must be acknowledged within 30 seconds and other alarms within 75 seconds. If there is no acknowledgement, the alarm is sent secondarily to an alarm center, e.g. SOS Alarm.

The guard carries a mobile unit for reception, delivery confirmation (acknowledgement) and reporting. There is also access to all other available forms of computer support. The mobile unit has a rapid and simple touchscreen interface and can be equipped with a uniquely identifiable code that identifies the unit concerned.

Figures 12 and 13 show a series of screenshots illustrating the course of events in the mobile unit.

Mobile alarm receiver 450 can, of course, be used in combination with all other possible variants of the monitoring system.

Figure 14 shows a possible function (integrated in the monitoring system) for the locking and opening of an alarm terminal for secure, remote configuration.

An alarm terminal can be configured for certain characteristics and may then not be modified without authorisation. At the same time, there is a need to allow, for example, service technicians to remotely service 410, i.e. it must be possible to connect remotely and alter settings, change programs, etc. However, this must never happen without the alarm terminal "owner" (400) actively assenting to this.

The problem can be solved by configuring the alarm terminal: to have, in its "normal condition", all interactive inputs and ports closed; and, to only allow outward communication (e.g. in connection with an alarm) to be initiated from within the terminal.

When remote access is necessary, the "owner" (400) uses an encrypted and digitally signed message to identify "himself/herself" to the alarm terminal. This

message approves and opens the terminal for remote access during a set period, e.g. 15 minutes.

On the expiry of this time, the terminal automatically closes the opened port. Using a new, encrypted and digitally signed message, it is possible to close the terminal manually.

The messages for opening and closing can be sent via another risk-free carrier such as SMS where a procedure in the terminal only accepts correctly coded and signed messages.

Here, it is also possible to use a dedicated TCP/IP network (620), which is a network held separate from the public internet network.

Figure 15 shows flow diagrams for three variants of how remote service, as per figure 14, can be effected. Of course, other variants are also possible.Remote service can be effected via a mobile network in the following way. Remote service 410 wishes to carry out remote service on central unit 120. Via mobile network 600, a connection is established with server 230. This checks whether customer 421 is open for remote service via mobile network 600. If the line is open for remote service, the connection is relayed, via mobile network 600 and wireless unit 131, to central unit 120. Remote service can now be effected. The opening for the connection is closed either manually on completion of remote service or automatically after 15 minutes.

Variant two shows a flow diagram for one way in which remote service can be effected via a fixed network. Remote service 410 wishes to carry out remote service on central unit 120. Via dedicated TCP/IP network 620, a connection is established with server 240. This checks whether customer 420 is open for remote serviceVia mobile network 600 and server 230. If the line is open for remote service, the connection is relayed (via dedicated TCP/IP network 620 and internet unit 134) to central unit 120. Remote service can now be effected. The opening for the connection is closed either manually on completion of remote service or automatically after 15 minutes.

Variant three shows a flow diagram for a second way in which remote service can be effected via a fixed network. Remote service 410 wishes to carry out remote service on central unit 120. Via dedicated TCP/IP network 620, a connection is established with server 240. This checks whether customer 420 is open for remote service via TCP/IP network 610 and server 230. If the line is open for remote service, the connection is relayed (via dedicated TCP/IP network 620 and internet unit 134) to central unit 120. Remote service can now be effected. The opening for the connection is closed either manually on completion of remote service or automatically after 15 minutes.

Figure 16 shows an example of a communication unit (130) and its constituent elements. The unit has interfaces for various communication possibilities and contains, for example, functions allowing it to work as both a fixed and a wireless IP terminal.

Redundancy and security are important functionalities in the monitoring system. The following describes one way of securing these.
It is judged that the solution set out below satisfies the requirements stipulated and broadly given concrete form in section 1.1.1. Seeking to exemplify the basic principle of the proposed solution, several example use cases are given in sections 1.6. and 1.7.

Definitions

Server cluster	A collection of servers in which failover support has been implemented. Failover means that if one server in the cluster goes down, one or more servers in the cluster will deal with its workload and jobs.
Alive server cluster	A server cluster in which, at one or several physical locations, failover support has been implemented for Alive functionality. Failover means that if one server in the cluster goes down, one or more servers in the cluster will deal with its responsibilities (terminals and alarm conditions).

1.1 Description of proposed system architecture

The proposed solution:

Places the decision on redundancy switching with the terminal. This reduces complexity throughout the system solution.

Modifies the server system solution with a dedicated (not redundant) configuration server and any number of Alive servers that handle terminals and alarm receivers. Each Alive server (230/240) can be located anywhere but requires IP access to all the other Alive servers in the Alive cluster as well as to the configuration server and the web access server (see figure 17).
With minimal (but necessary) state information so that any server can quickly take over a terminal if it chooses to change the IP address for communication in towards the system, each Alive server is continuously synchronised with each other.

Simplifies the database structure in Alive servers 230/240 so that it only handles state changes and does not store log information/statistical information which is not used. This involves removing the terminal log, terminal alarm log and system log from MSATB and ACTB, thereby minimising the amount of data that is written for log purposes in the database. Can be written to a file so that, for troubleshooting purposes, all log files from all Alive servers can be compiled for database analysis.

In this solution, the configuration server does not need to be up for the system to work. However, it does need to be up for any modifications in the configuration of the system or an individual terminal.

The basic idea in the proposal is the introduction of a list of Alive server IP addresses. This is loaded into the terminals and the choice of live server is then left to each terminal. Compared with the present terminal, the difference is that, instead of a unique configuration IP address and a unique Alive server address, there is a list of possible IP addresses that can be used for communication with the system. A terminal can use any of the IP addresses for communication and the decision to change servers when contact is not made with a server is taken by the terminal itself. These servers are used both for downloading configurations and for Alive messages, i.e. there is no longer any distinction between the various messages.

Implementation at the terminal is easy. When communication fails with one server (an IP number), the terminal passes to the next IP number in its list. The terminal is initially supplied with a factory configured list. However, using new, remotely loaded software, it can be modified to support new lists.

For this to work, all the servers in the specific terminal's "area of responsibility" (the list) must know which server is currently handling the terminal.

1.1.1 Alive server 230/240

Below, there is an overview of the proposed modification of the Alive server.

Figure 18 shows that large parts of the present Alive server implementation can be retained. The new function that synchronises all the Alive servers in the cluster is described in more detail in section 1.1.2.

The functions set out in figure 18 are described briefly in the table below.

Function	*Comments*
Terminal communication Alive TCP service Alive UDP service Alive SMS service	All present-day communication services (TCP/UDP/SMS) require no major architectural changes, but do require modification of: - application protocol handling for the changes in configuration messages to support the new handling of the Alive server list instead of the fixed configuration and Alive server in the present system solution.
Alarm handler Alive push alarm service Alive multicom online service Alive multicom wireless service	With the reservation that certain adaptation may be needed when implementation details become clearer, it is judged that all present alarm communication services can be used without modification.
Other Alive MSMQ reader Alive scheduler service	With the reservation that the present Alive/alarm analysis is used, it is judged that these can be used without adaptation. If Alive/alarm analysis is placed in its own service, parts of MSMQ reader can be transferred into this service.
Alive/alarm analysis	There are two possibilities here: - to keep the present SQL 2000 server job Alive analysis. However, this would require a number of changes to handle the new state parameters for active server - to transfer the job Alive analysis function to a Windows service. Because only state changes would be saved to the database, and other information would be written to log files, this would result in less writing/reading in the database.
Alive database Terminal state Terminal configuration System configuration Terminal and server log	There are also two possibilities here: - to extend the present table structure that mixes various parts. This would require less recoding but, because the complexity and maintenance would become increasingly extensive, would result in more work. - to modify (reduce) the database structure for a complete restructuring of the database contents in the 4 logical parts (because it has a linearly alternating character, it could also be considered whether the terminal and server log database should be separated).
Server synchronisation service	New service that is used for synchronising databases in the Alive server cluster and

	configuration server and the web access server. The logic in the proposed redundancy handling on the server side.
Data push	New service that is used to enable unique configuration, on an individual terminal basis, for the pushing of selected data, e.g. Alive position, fleet, etc., in a similar manner to the present Wireless Alarm Push (SOS V3). This is a basic component for customers who require access to current positions and fleet data.
Log files Terminal log System log Error log	File based log that could be handled, if so required, by a new function, "Log Collector" in the configuration server and the web access server.

The choice still remains as to whether each logical server should be divided between two physical servers (one communication server and one database server).

The arguments for placing everything in a single physical server are:

- Cost saving (i.e. around half the amount per Alive server). Savings on server hardware and an MS 2000 server license.
- Simplified maintenance. Because the number of physical servers in the system is reduced, installation/upgrading/backup and system maintenance are simplified.

The arguments in favour of continuing to use the "distributed solution" between server front end/back end (communication server/database server) are:

- Hardware/configuration optimisation. The communication server and the database server can be optimised as regards hardware and configuration.
- Because the back-end IP is not exposed (depending on the network solution used in the implementation), greater possibilities for securing the system from unauthorised access/holes.
- Because each node in the cluster would be more efficient, the number of nodes in the cluster could be kept down. This would benefit synchronisation traffic, which increases with the number of cluster nodes in the system.

Given the reasoning above, the argument for continuing with front-end/back-end distribution seems to have the advantage where finance and maintenance/operation time permit.

1.1.2 Synchronisation service

The synchronisation of terminal **state data** in each Alive server is necessary to enable all the servers in the system to handle any terminal that chooses to change server for its communication.

Furthermore, so that the configuration of **system parameters** and unique **terminal parameters** (which is effected in the configuration server database) can be updated and reflected in the Alive servers, there must also be a synchronisation function for these parameters. The difference between this and the preceding is that, here, the configuration server is the master and the only server that can implement changes of the parameters in the system.

To handle the two functionality requirements given above, a synchronisation service (installed on all Alive servers, the configuration server and the web access server in the system) is implemented. The synchronisation service establishes an outgoing TCP session (and keeps the session live) with all the servers in the system and implements a listener for incoming TCP sessions from all the servers in the system.

For the synchronisation service to work satisfactorily, all the servers in the system must have good time synchronisation. Good time synchronisation is a critical parameter for the synchronisation service when it has to determine when/if state updating shall take place.

Figure 19 shows that each Alive server in the cluster has one (heartbeat monitored) client session with each Alive server in the cluster (apart from itself). This session is used to update terminal state changes.

In addition, each Alive server has one client session with the configuration server and the web access server, which, so that it can present terminal status via MSATB or ACTB, also receives terminal state changes.

The configuration server also has a client session with each Alive server in the cluster. The session is used to send configuration updates (these can only be sent from the configuration server) and terminal control commands.

Figure 20

This synchronisation service, Terminal State Synchronisation Service, is to comprise:

A "**state update send**" function that is responsible for sending out "update terminal state" messages (for terminals whose state changes in this server) to all other servers in the redundant cluster.

The "state update send" function sets up an outgoing TCP session (which is heartbeat monitored if no state updating is necessary) with each A-dressable server's "state update receive" in the Alive cluster. A configuration file (or equivalent) that gives:
- IP addresses of Alive servers
- The IP address of the configuration server
- The IP address of the web access server

A "**state update receive**" function that is responsible for listening and setting up outgoing sessions with configured Alive servers in the redundant cluster.

The "state update receive" function is heartbeat monitored and is implemented as a TCP listener that approves sessions from all addressable (configured) servers' "state update send" in the Alive cluster.

A "**state update logic**" function that implements the logic in the synchronisation service and is the function that receives:

1) Indication from Alive analysis when a terminal's state changes and generates a message out to the "state update send" function. The logic function is also responsible for keeping track of whether state updating fails towards one or more servers. When a session can be established, it repeats state information for this terminal to the server(s).

2) Incoming heartbeat messages from all servers in the server solution.

3) Incoming "update terminal state" messages from other servers. It then updates the database as per the received information.

4) Incoming "update terminal configuration" and "update system configuration" messages from the configuration server. It then updates the database as per the received information.

5) Incoming "send terminal message" messages from the configuration server. It ensures that the message is sent from an active server.

The logic function is also responsible for re-establishing connections with all servers in the cluster and the configuration server and the web access server.

Communication between "state update send" clients and "state update receive" servers is to use TCP (maintained session) and a new, specially adapted application protocol that is used for information exchanges and which offers the following message types:

Request/Response	Description
Heartbeat	Used to monitor TCP sessions. Always sent every x seconds (where x is configurable). To make the message more useable, it should include parameters such as, for example, the current terminal load, server operation state, etc.
Update terminal state	Used by the Alive server to update terminal state in all the servers in the system.
Update terminal configuration	Used by the configuration server to update terminal configuration in all the servers in the system.
Update system configuration	Used by the configuration server to update system configuration in all the servers in the system.
Update terminal server log	Used by all the servers in the system to update necessary (classified as so important that synchronisation is necessary) log information. Unlike the present "terminal log/terminal alarm/system log" level, only necessary information is updated. At an

	estimate, only 2% of present log information requires synchronisation.
Send terminal message	Used by the configuration server to send terminal specific messages triggered by an external function (MSATB).

1.1.3 Terminal and system configuration information

The following information is classified as terminal and/or system configuration information and can only be modified in the configuration server. This server's synchronisation service then ensures that changes are distributed to the other servers in the system.

Global system parameters:

Service levels
Roaming lists
Remote download software versions
Alarm center configurations
Alive server list (Remark: New list handling vital for redundancy)

Unique parameters for each terminal in the system:

Installation parameters (MAC, IMEI, SIMSERIAL, MSISDN)
Configured service level
Configured roaming list
Configured software version
Alarm port settings (activation and input type, connection to alarm center)
MLAN settings (IP settings, port configuration)
NMEA push configuration

New configurations and updating/modification of existing configuration information in the configuration server is effected, as earlier, using the existing MSATB API. Other than the addition of Alive server list handling, this API requires no modification as regards configuration parts. Expansion of the API may also be required for redundancy checking and monitoring/operating statistics in respect of such checking.

As in the present solution, MSAWEB/MSATB is used for changing a terminal's configuration and/or the system configuration. All changes are made in the configuration server's database. The synchronisation service then ensures that the change is sent out and updated in all Alive servers in the system.

1.1.4 Terminal state information

The information below is classified as terminal state information and must be synchronised each time the state changes. It is grouped in the following classes:

- Global state change time
- Connection status
- Outbound message state
- Alarm state (alarm input status, alarm code, alarm time, alarm center rec. ID)
- Crypto state (Key ID/Secret)
- Responsible server

<GlobalStateChangeTime> indicates the latest date/time when any of the state parameters changed. Each state class also has its own update time parameter (class state change time) for this purpose.

<ConnectionStatus> indicates the terminal's connection status (Alive analysis) – refer to the table below. This status information already exists in the present database.

<OutboundMessageState> is a modified version of the present outgoing message queue. The modification adds the state that a certain type of message is to be sent until the corresponding delivery confirmation is received from the terminal. This is to enable handling of configuration updates, new terminal software downloads, etc in the new redundancy solution without having to synchronise the present message queue in the database.

<AlarmState> indicates the alarm state for the eight alarm inputs and the connection alarm.

<CryptoKeyState> always contains the latest valid encryption information in the form of "key ID" and "crypto secret".

<ResponsibleServer> indicates which server in the cluster is responsible for the terminal in question.

Each and every one of these groups of messages has its own time column in which <ClassStateChangeTime> is logged so that the synchronisation service's logic can process decisions on current terminal state where several different states have arisen owing to the error conditions that can occur when the synchronisation service between different servers has not worked.

Each terminal is identified by an individual ID. This is the unique key for each terminal in the database. Each change in a unique state parameter is logged with the current time (this requiring that all the servers in the system are time synchronised) so that, in special cases, it is possible to handle the error conditions that can arise. Each state parameter can have the values set out below.

Parameter	*State*	*Comments*
<ConnectionStatus> <Status> <StateChangeTime>	*In Sync [1]* *Inactivated [4]* *Activated [5]* *Alive stopped [6]* *Out of sync. [1001]* *Alarm out of sync. [2002]*	In present database
<OutboundMessageState> <Status> <StateChangeTime>	A combined list of message types in a queue that is to be sent to the terminal: *Request download of new configuration* *Request file download (nvparam.nv, natdb.nv, etc.)* *Request report of software version* *Request key change*	New state variable replacing the message queue table in the present database In present database

<AlarmState> [9] {	0..9 Alarm inputs, where 0 = connection alarm
<AlarmInputStatus>	
<AlarmCode>	
<AlarmTime>	
<AlarmCenterRecID>	
<StateChangeTime>	
}	
<AlarmInputStatus>	*Waiting for alarm on input* *Alarm reset acknowledged from alarm center* *Alarm out of sync. reset, acknowledged from alarm center* *Alarm received from terminal* *Alarm acknowledge received from alarm center* *Alarm acknowledge sent to terminal* *Alarm reset received from terminal*
<AlarmCode>	*0,1,2,..,256,512,..32768* States alarm input where 0,256..32768 is used
<AlarmTime>	*YYYY-MM-DD hh:mm:ss.sss* (UTC date/time format)
<AlarmCenterReceiverId>	1..N
<CryptoState>	*Key ID*
<KeyID>	*Crypto key*
<CryptoKey>	
<StateChangeTime>	
<ResponsibleServer>	<IP>
<ServerIP>	
<StateChangeTime>	"xxx.xxx.xxx.xxx" or 0.0.0.0 (IP address of the Alive server responsible for the terminal)

1.1.5 Configuration server

Below, there is an overview of the proposed, new, configuration server function. This exists as a logic part of the present system, but it cannot be separated (which is possible in the new system solution) and it does not handle terminal communication to this server.

Figure 21 shows that the configuration server uses exactly the same database as Alive server, but with the difference that the configuration server is the (sole) unit creating and modifying the content of the terminal and system configuration. Furthermore, it receives a copy of terminal state through passive listening of the terminal state updates received by the server synchronisation service. The terminal

state database is used to enable the system administrator to get all current terminal states via MSATB. Unlike the Alive server, the configuration server has no communication interface with terminals and alarm receivers. Instead, it offers the MSATB API for management of the system's terminals.

So that the new operator functions supplied in Server v2.6.x can be handled in the new system architecture, it must be possible for the server synchronisation service to cope with the configuration server's ability to send terminal unique messages to the server in point. This is done with the "send terminal message request" type message.

The table below briefly sets out the functions required in the configuration server shown in figure 21.

Function	*Comments*
MSATB	The present M2Mserver Administration Toolbox API, which must be expanded to support the new parameter (server cluster addresses) that, for redundancy management, is downloaded to each terminal. It will also be necessary to document the reduction in the amount of log information that is presented. In all probability, the API is not affected, only the amount of data that is returned and the permitted values in the search filter in calls to terminal and system logs.
Server synchronisation service	Synchronisation of databases in the Alive servers in the cluster, the configuration server and the web access server.
Alive database Terminal state Terminal configuration System configuration Terminal and server log	The same database as previously described for Alive server.

1.1.6 Web access server

Below, there is an overview of the proposed, new, web access server function. This exists as a logic part of the present system, but it cannot be separated (which is possible in the new system solution). It is being completely restructured for customer access. At the same time, because no configuration is possible from this server, security is increased.

Figure 22 shows that the web access server uses exactly the same database as Alive server (with the addition of a new table for handing terminal history). It receives a current copy of terminal state through passive listening of the terminal

state updates received by the server synchronisation service. From the configuration server, it also receives a current copy of the applicable terminal configuration.

Unlike the present Alive server, the web access server has no communication interface with terminals and alarm receivers. For customer management of the system's terminals, it offers solely the ACTB API.

Retaining the possibility of downloading current position and terminal log history via ACTB requires the following new developments:

A push service, MSA Wireless Data Push, for non-alarm based data such as, for example, Alive position, fleet and, where relevant, transparent data from the terminal's series port). This push service can also be connected to push the information directly to the customer. However, it must always be configured to push configured data (e.g. Alive position) to the web access server and the service described immediately below.

A unique table in the Alive data database that is updated by a new service, MSA Wireless Data Push receiver, that listens to all push services from the Alive servers in the cluster.

The table below briefly sets out the functions required in the configuration server shown in figure 21.

Function	*Comments*
ACTB	Present M2Mserver Alive Client Toolbox API It will be necessary to document the reduction in the amount of log information that is presented. In all probability, the API is not affected, only the amount of data that is returned and the permitted values in the search filter in calls to the terminal log.
Server synchronisation service	Synchronisation of databases in the Alive servers in the cluster, the configuration server and the web access server.
Alive database Terminal state Terminal configuration System configuration Terminal and server log Terminal history log	Same database as previously described for Alive server, but with the addition of "terminal history log". This is used to present current and historic Alive position/fleet to customers

1.1.7 Online alarm handling

The Online system and the connection of Wireless to this system has shown that the proposed solution can be handled when multiple node addresses are allowed in the system. The solution is then that all the servers in an Alive server cluster use the same node address in outgoing communication to the Online network (see figure 23).

One new function that has to be handled is the fact that alarm acknowledgements from the Online network will be routed to any Alive server in the cluster irrespective of from which server the alarm was sent. This means that the Online service/alarm analysis at the server must also handle incoming delivery confirmations to terminals for which the server is not responsible. The terminal's state change is updated via the synchronisation service but, as the terminal is still handled by the active server, with no change in the "responsible server" state parameter.

5.1.8 Alarm handling in SOS Access version 2 XML

Alarm handling in servers for terminals configured to send alarms over the SOS V2 interface is not affected by the redundancy solution. This is because the interface requires delivery confirmation within 10 seconds of a session being established. Consequently, delivery confirmation will always come into the sender (the active server).

All servers are thus configured to send SOS V2 alarm messages over TCP/IP to the same IP. SOS only has to open the firewall for the IP addresses of all the Alive servers in the cluster. This is because the interface requires delivery confirmation within 10 seconds of a session being established. Consequently, delivery confirmation will always come into the sender (the active server).

5.1.9 Alarm handling in MSA Wireless Push (~SOS V3)

Alarm handling in servers for terminals configured to send alarms over the MSA Wireless Push interface is not affected by the redundancy solution. This is because all the servers are configured to send Wireless Push alarm messages over TCP/IP. Alarm acknowledgement always takes place in the established session and, consequently, a transmitted alarm will always be sent back to the active server.

Unfortunately, the present implementation of alarm receivers' is affected as the receivers must:

1) Handle TCP sessions coming in from all the Alive servers in the cluster. Thus, for all Alive servers in the cluster, openings must also be made in their firewalls.

2) Implement outgoing (client) heartbeat sessions with all Alive servers in the cluster.

1.1.10 SMS proxy server

Is not affected by the redundancy solution.

1.1.11 Terminal specific comments

The modifications needed in the terminal application are not judged to be so extensive and the algorithm for redundancy switching can be described, in a simplified manner, as follows:

Loaded at manufacture, the terminal has a default list of permitted Alive servers (IP addresses) in the application.

To fetch the time (if it does not get this from the optional GPS module) the terminal initially contacts the first server in this list. If contact is not established, a cursor moves over the list of active servers and the terminal tries to fetch the time from the new IP address.

When the terminal has fetched the time, it uses the active server (cursor in the file) for all Alive related IP communication, i.e.:

Key exchanges (RSA encrypted)
Alive messages (AES encrypted)
Time synchronisation messages (time client).

Any SMS messages that are sent also always use the active server IP.

If the terminal's TCP/IP transmission of any of the above messages fails, the terminal tries a further N times (N is a configurable parameter, as a suggestion N = 1). The terminal then moves one step forward in the server IP list and uses this IP for all Alive related communication. If the cursor is at the end of the list, it starts again with the first IP address in the list.

Before activating SMS transmission, a terminal that has been configured to use SMS as its secondary carrier waits until, let us say, its first attempt, to try a new IP address using GPRS. As previously noted, the SMS transmission uses the IP address currently indicated in the list.

The file list cursor is to be saved in flash memory. Thus, if the terminal is deprived of power or restarted, it will start up again against the most recently used server IP. This is to avoid generating unnecessary synchronisation traffic in the Alive cluster.

1.2 Handling of load balancing

Depending on needs and requirements, because every server will be able to send a "switch server" message to individual terminals, this system solution enables various types of load balancing.

Load balancing can thus be implemented in one or more of the following ways:

- The number of terminals to be handled by a server is exceeded and the server chooses to force (randomly or in a predefined manner) terminals to switch to new servers.

- Load monitoring information can be built into the heartbeat synchronisation message that is sent to all servers. This would permit logic in each server to dynamically distribute the load by forcing (randomly or in a predefined manner) terminals to servers with loads that are reported as low.

To enable a server to force a terminal to switch servers, a new message is to be introduced in the application protocol sent from server to terminal. It will state the new IP address the terminal is to switch to.

1.3 Error monitoring and automatic redundancy switching

A new logic function that can be introduced with this solution embodies an improved monitoring function in respect of Alive server services and log files. The new function can automatically decide to close a server when it shows errors and, at the same time, load balance the terminals that used this server (i.e. split the load amongst the other servers in the cluster). The error can then be remedied "offline" without affecting the functionality of the terminals.

As previously, this function can be implemented by using a third-party program. It is most simply implemented by closing all relevant services on the server in question when the monitoring system triggers a serious error state. This closure will lead to all terminals and alarm receivers that were operating towards the affected server being taken over by another server in the cluster.

Implementation requires both a review/analysis of the events and also documentation of the Alive server log messages triggering the events. In certain cases, implementation may require stored procedures and changes/improved error messages from services.

1.4 Migration of existing TOR terminals

To enable the introduction of this redundancy solution, it must be possible to update existing TOR terminals to new software that supports the new redundancy solution (the update would, of course, also have to be implemented). Although no details have been specified, it is not judged that this presents a problem. The amount of new code required to handle redundancy switching in terminals is

relatively small. The following is one way in which migration of existing terminals to the new redundancy solution can be implemented:

1. The new, redundant server architecture is installed and put into operation parallel with the present Alive servers handling the existing terminals.
2. Terminal data and configuration data is exported (manually) from the present Alive servers. So that no state changes occur during the migration period, which it should be possible to keep relatively short, the Alive server system is stopped.
3. A script especially adapted for migration purposes is used for the import of the old format. This brings the entire server architecture into operation. One of the Alive servers in the cluster is mapped as the "old" Alive server.
4. New software that supports the new Alive server list and redundancy handling in the terminals is downloaded remotely.
5. The new redundancy solution is now live.

1.5 Alarm analysis in a redundant system

It is judged that, if so stipulated, all four proposed alarm analysis types (see below) can be handled in the proposed redundancy solution. However, as only one of the four variants (loss of ISP) is implemented in the present server software, this would require further, newly developed software.

Connection alarm analysis based on network information from the operator (cell ID/local area).

Connection alarm analysis based on information from reference terminals.

Configuration of terminals to be included in analyses.

Filtering out of connection alarms caused by loss of ISP.

1.6 Use cases – normal operation

1.6.1 Terminal start-up

The steps in a normal seuence for the start-up of a terminal in an existing Alive server cluster system are as set out below.

Import of terminal data
Using a terminal import tool, the terminal's installation parameters are imported (from the production database) to the configuration server.

Configuration of the terminal in the configuration server
Other terminal configuration parameters (service level, MLAN settings, alarm port settings, NMEA push, roaming list) are set up in the configuration server. The Alive cluster to be used by the terminal in question is also allotted here. On completion, configuration is confirmed, terminal state is activated and the configuration server's synchronisation service sends an

update of the terminal configuration to all the Alive servers in the cluster. All these servers are set as "not responsible" for the terminal. More precisely, to indicate that in the present condition no server is responsible for the terminal, the "responsible server" field is set to <0.0.0.0>;1970-01-01 00:00.00".

Opening for initial key exchange
At a given point in time (e.g. triggered by a Prisma/MSA web operator), the terminal is opened (from the configuration server) for the initial key exchange. The configuration server mirrors this configuration change to all "included" servers. For safety reasons, this window, which is the one where the terminal is allowed to effect a key exchange, should be kept as short as possible.

The terminal fetches the time
For the terminal to be able to communicate with the servers, a correct time must be used in the encryption store. Thus, when the terminal has started and established GPRS, it begins by fetching the time using the simplified SNTP protocol that has been implemented. Here, the terminal chooses to fetch the time from the first server in its pre-configured list, (e.g. A1). If the terminal is not successful in fetching the time from this server, it goes on to the next server in the list. Alternatively, if the terminal is equipped with a GPS module, it can get the time from this. Here, it does not contact the server for time synchronisation. To fetch the time, the terminal does NOT need to be configured in the system. This is because the time server in each Alive server listens to time requests from any client and does not require any authentication of the terminal.

The terminal begins initial key exchange
When the terminal has fetched the time, it contacts the active server (the first in the list or the one from which it fetched the time) and sends an RSA encrypted key exchange message. If the terminal is started (from the configuration server) before it has been opened for key exchange, the server it contacts is not able to identify the terminal as valid. This is logged as an error in the system log/event log. After N (configurable parameters) failed transmissions of the application message, the terminal decides to move on to the next server in the pre-configured list. This procedure repeats for as long as there is no opening for key exchange.

New key and key ID for the terminal
The first server receiving the message from the terminal checks the "responsible server" terminal state parameter. If this is set to "0.0.0.0", the terminal's message is accepted and the server takes responsibility for the terminal by generating a key/key ID. This is sent to the terminal, the terminal state is updated with the new values and "responsible server" is set to the server's IP address.
The state change is triggered up to the synchronisation service. Using all the servers configured for use with the terminal, the synchronisation service then sends the new terminal state to all servers in the Alive cluster, the configuration server and the web access server.

Downloading the configuration
The terminal then continues to send messages to the active server. As it initially lacks a configuration file, it begins by downloading the configuration.
The terminal is now active and in operation.

1.6.2 Modifying terminal configuration

The steps in a normal sequence for modifying a terminal's configuration are as set out below.

Modifying a terminal's configuration
As with the present system, the operator modifies terminal configuration (via MSAWEB or other interface) by accessing MSATB. If the terminal configuration modification is accepted, the information is updated in the configuration server.

Updating the configuration in the Alive cluster

The configuration server's synchronisation service sends a configuration modification update to all the Alive servers. If the configuration modification requires the terminal to be updated (e.g. downloading a configuration or a message file), the synchronisation service also triggers this by updating the terminal state with the relevant parameter in the <OutboundMessageState> state class.

The Alive server sends messages to the terminal
When the server responsible for the terminal in question has received the state change, it begins (in the absence of prioritisations) sending messages to the terminal. The messages relate to the requested configuration modification (e.g. downloading a new configuration). If, at this point, the terminal experiences problems connecting to the active server, it passes on to the next, which also has the same state parameter activated (i.e. to send, for example, a download new configuration message to the terminal).

The terminal downloads the new configuration:
When the terminal has downloaded the requested configuration modification, the responsible server updates the state change (removes the active parameter in the <OutboundMessageState> state class). This is synchronised out to all Alive servers, the configuration server and the web access server.

1.6.3 Operator access

As in today's system, the operator logs in via MSAWEB, the Prisma customer management device or any other operator interface that uses MSATB to communicate with the configuration server.

The difference in the new redundant system is that the amount of information in the system log and the terminal log (and perhaps also the terminal alarm log) is less. Otherwise, most things are exactly as in the present MSAWEB/MSATB system. To enable configuration of the Alive server list, an addition will have to be made.

REMARK: Because certain log functions are no longer supported in the new API and there is a new addition, the Prisma application, which developed services for MSATB V2.5.2 (= V2.6.1), must update its applications.

1.6.4 Customer access

The customer logs into the ACTB web service in the web access server. All the data presented to the customer comes from a local, synchronised database in the web access server.

REMARK: Customers who have developed services for ACTB V2.5.2 (= V2.6.1) must update their applications. However, as the only effects on the API are the amount of data that can be returned and a reduction in certain input parameters, it is not judged that updating will require extensive work.

1.6.5 Adding and starting up a new server

The steps in a normal sequence for starting up a new server are as set out below.

Installation of Alive server software
The authorised person installs all the software on the Alive server. The system installs basic configurations and a database without terminal and system data. In the database, a table (system config.) indicating the database's last configuration update is set to 1970-01-01 00:00:00. This indicates that the database lacks all configuration data. In this condition, all the communication services are not activated on the new Alive server.

Updating of system and terminal configurations in the new server
The new server's IP address is included in the list of Alive servers in the configuration server's configuration of the synchronisation service. When the configuration server's synchronisation service starts to send <Heartbeat> messages from the configuration server to the new Alive server, the answer from this server will include the time/data of the latest configuration state. Consequently, the configuration server's synchronisation service discovers that all configuration data must be updated in the new server. The configuration server thus updates the new Alive server using the <UpdateSystemConfiguration> and <UpdateTerminalConfiguration> messages. This fills the new server's database with the current configuration data. What the server now lacks is state information for all the terminals.

Updating the synchronisation services in the Alive cluster
All the synchronisation services in the existing Alive server cluster are updated with the new server's IP address. This leads to all the servers establishing a client session with the new server. They begin by sending <Heartbeat> messages. The new server responds with a <Heartbeat> response message. This has a parameter stating the time of the latest update of state data from the active server (as the new server lacks earlier state information, this will be 1970-01-01 00:00:00 in all cases). All the servers in the cluster receive heartbeat responses and, by sending an <UpdateTerminalState> message, update the state of all terminals whose date/time for state change is later than that indicated. The new server receives <UpdateTerminalState> messages from all the servers in the cluster and, on the basis of the received data, updates its database.

Start of communication services on the new server
Next, all communication services on the new server start. This is indicated in the <Heartbeat> response message to the other servers in the cluster by the "operations state" parameter now being set to "running".
If the "spread equal" load balancing function is activated, the other servers in the cluster discover that the new server has no load. To even out the load balance resulting from the insertion of the new server, they will gradually transfer terminals to the new server. If load balancing is not to take place, the new server is ready to handle terminals should there be problems with any other server in the cluster.

Updating the Alive server list
Finally, the system's Alive server list is updated on the configuration server. This leads to all the terminals downloading a new Alive server list. Next, all communication services on the new server start. This is indicated in the <Heartbeat> response message to the other servers in the cluster by the "operations state" parameter now being set to "running". This enables the terminals to switch, on redundancy, to the new server.

1.6.6 Alarm sequence with Online as the alarm receiver

The steps in a normal alarm sequence with Online as the alarm receiver are as set out below.

Alarm at a terminal input

The terminal in question sends an alarm message to Server A. The alarm message is received by server A, which updates the terminal's state in the database. This triggers synchronised update with all the other servers in the cluster.

The alarm is sent to the Online network
Via the series port, Server A sends an alarm to the Online network.

The Online network sends delivery confirmation of the alarm to server B
Because all the Alive servers in the Alive cluster are identified by the same Online node number, the server to which the delivery confirmation is "routed" is entirely random (routing algorithm in the Online network). The server (server B) receiving the confirmation will, if not itself responsible for the terminal, handle the confirmation in the expected manner.

The synchronisation service updates server A's state
Using the <UpdateTerminalState> message, the synchronisation service on server B updates all the servers in the cluster (server A included therein) with the state change.

Server A returns a delivery confirmation to the terminal
Server A assimilates the state change and sends an alarm confirmation back to the terminal.

1.6.7 Taking an Alive server out of operation

The steps in a normal sequence for taking an Alive server out of cluster operation are as set out below.

Stopping of server services
The operator stops the communication services (TCP/UPD/SMS) and the alarm services on the server in question. This will result, at the next message, in all the terminals that access this server switching (redundancy switching) to the subsequent server on the list.

Verification of redundancy switching
On the terminal state list in any server (or via the interface in MSATB), the operator verifies that none of the configured terminals are any longer handled by the stopped server. If the latter is not the case, the operator must wait until it is (can, of course, be automated if this is desirable).

Updating the Alive server list
If it is only a short stop in the server's operation, it is probably unnecessary to update the Alive server list. For longer stops, the operator updates the Alive server list in the configuration server. This configuration change is mirrored to all the Alive servers in the system. These ensure that it is downloaded to all the terminals. All synchronisation services are also updated with this information and no communications will be sent to the stopped server.

1.6.8 Taking an Alive server back into operation

The steps in a normal sequence when an Alive server is taken into operation after a short period of maintenance without the Alive server list having been updated (by the removal of the server from the list) are as set out below.

Start of synchronisation services on the server
The operator starts the synchronisation service on the server. The configuration server begins to update the server. Detecting from the <Heartbeat> response message that the time of the last configuration and terminal state update is far earlier than the latest change (provided this took place while the server was out of operation), all the other servers in the cluster also contribute to this updating.

Start of communication services on the server

The operator waits until he/she has received an approved status (from the synchronisation service) for the server that has been put back into operation. This is given by all the servers in the cluster having updated the server with current state information and the configuration server having updated any configuration changes. When the approved status has been received, the operator starts all communication services on the server. The server is now back in the redundancy cluster and can handle redundancy switching by the terminals in the system. If load balancing is activated, other servers can detect (from the <Heartbeat> information) that this server is unused and, in accordance with the algorithm implemented for this type of load balancing, transfer some of their terminals to the unused server.

1.7 Use cases – error handling

1.7.1 Failure in a terminal's TCP connect with a server

Below, step by step, is the course of events when a terminal's message to an Alive server fails. This can happen when:

Access to the server is disrupted (ISP problems, firewalls, network faults).

Problems in the server environment prevent an incoming TCP session being established

All errors (besides those given above) causing failure in the TCP connect to the active server have the following pattern:

The terminal sets up a TCP session
This happens when the terminal fetches the time or sends encrypted Alive messages to the server. For each encrypted message that is to be sent, the terminal takes the active IP address from the list (in its file system) of available servers in the Alive cluster. The terminal attempts to establish a TCP session with the given address, but fails when, due to any of the above-given error conditions, the server does not reply.

The terminal changes active server in its list
The terminal goes one step forward (from server A to server B) in the list of permitted Alive servers and seeks to establish a TCP session with the indicated server.

The new server receives an encrypted message from the terminal
The new server (server B) with which the terminal is communicating decrypts the message. However, as the "served by date" state parameter is "server A" and the "served start time" state parameter indicates the "historic" time when server A took "over" the terminal, server B notes that the terminal was not previously its responsibility. For this terminal, server B updates, in its database, the "served by" state to "server B" and the "served start time" to <current UTC time>. The synchronisation service ensures that this state change is mirrored to all the servers in the Alive system (the Alive server cluster, the configuration server and the web access server). Server B has now taken over responsibility for the terminal in question. If the synchronisation service on server B does not have contact with the synchronisation service on server A, the re-synchronisation service in server B will ensure that server A is updated with this state when server A once again establishes a client session with server B.

1.7.2 Synchronisation service in server X loses contact with all other servers

What happens when a server in the cluster loses synchronisation service contact with all other servers in the cluster, e.g. if the synchronisation service is stopped or fails, or if internet access between the servers in the system is lost? There are two choices in this situation:

1) The server automatically stops all communication services. This results in all the terminals that access this server switching (redundancy switching) to the next server on the list.

2) The server continues to handle all its terminals. This means that alarms/resets coming in from the Online system are not routed back, via the synchronisation service, to the server.

Choice 1 is the recommended alternative. A problem could indeed arise if only the synchronisation services on all the servers in the system were disrupted. This would result in a completely functional cluster solution suddenly ceasing to work as all servers close down their communication services. This does mean that the synchronisation service, as an application, is a sort of "single-point-of-failure." However, it is judged unlikely that such a fault in the application would not be discovered at implementation and validation.

1.7.3 Synchronisation service in server X loses contact with another server

What happens when a server in the cluster loses synchronisation service contact with one of the other servers in the cluster, e.g. if the synchronisation service on the other server is stopped or fails, or if internet access between the servers in the system is lost? The synchronisation service seeks unremittingly to re-establish the session with the other server. When it is re-established, the <Heartbeat> function between the two servers ensures that all state changes that occurred during the break are updated between the two.

1.7.4 The data push service loses contact with the web access server

In this situation, customers are not able to fetch current terminal positions and are without log history throughout the break. If this is considered critical, a function can certainly be implemented to fetch data and import it from the log files created locally on every Alive server in the cluster. However, this cannot be carried out if customers do not request it. Depending on how important this function is

considered, redundancy over two web access servers can be introduced. This spreads the risk of such a situation arising.

1.8 Identified software changes

To give an indication of the work required to implement the proposed redundancy solution, the two tables below give an overview that weights the extent of the changes (or newly developed code). Most of the changes are in the newly developed functions in the server software. This, in itself, is good – validation is simplified when the amount of code and functions that otherwise have to be changed can be held to a minimum.

Terminal software changes

Activity	Extent
Handling of the Alive server list (file in the file system)	Small
Handling of changed application protocol for Alive server list handling	Small
Handling of algorithm at redundancy switching and handling of SMS as secondary carrier	Small

Server software changes

Component	Activity	Extent
Communication services	Handling of new application protocol messages and configuration file for the Alive server list.	Small
Alarm handling services	No or little change. Triggering to synchronisation service and any necessary routing of Online delivery confirmations.	Small
Analyses	Transfer of job analysis to own service, but still through calls to stored procedures. Enables synchronisation between this service and the synchronisation service (this is judged necessary).	Small to moderate
Alive database	Opt. 1) Expansion of present structure to support redundancy requirements.	Moderate to large
	Opt. 2) Review of database structure with complete restructuring of configuration and state information.	Moderate to large
		Moderate

	In both cases, terminal logs and storing of Alive messages will have to be done away with. Writing will now be to a log file.	
Synchronisation service	New function that is required to handle the proposed solution.	Very large
MSA Wireless Data Push service	Modification of present alarm push service.	Small to moderate
Data push receiver	To enable updating of terminal history in the web access server.	Moderate
ACTB/ACW	Modification for limitations of API.	Small
MSATB/MSAWEB	Modification for handling of redundancy parameters and reduced log information.	Moderate
Improved alarm handling	A basic condition for the simple and fuss-free operation of this system is that a review of the error logs created by the system is gone through iteratively with operating staff during the implementation period.	Large

PATENT CLAIMS

1. Arrangement of units to form a monitoring system comprising a number of local alarm systems and a number of different monitoring stations, the whole being **characterised by**: the local alarm systems each having the task of showing, at their particular locations, the statuses of the various devices in the local system, such indication being achieved using, for example, at least one surveillance camera; the local alarm systems each having the means to send messages or other information about the statuses of the devices in the local alarm system to one or more monitoring stations, it being possible to send said messages or information over a number of different communication channels, this increasing the certainty of the messages/information reaching the intended destination; it being possible to equip each monitoring station with not only means for the remote checking of the extent to which each local alarm system is able to send messages, but also means for the remote configuration of the local alarm systems and means for directing the information streams from the local alarm systems.

2. Arrangement of units to form a monitoring system as per patent claim 1, **characterised by** the possibility of the communication channels being fixed telephony, TCP/IP and mobile communication, either separately or in combination.

3. Arrangement of units to form a monitoring system as per patent claim 1 or 2, said arrangement including one or more monitoring stations, at least one of which has a monitoring device for wireless units, the whole being **characterised by** said monitoring device sending messages, over a mobile network, to one or more wireless units situated within a local alarm system, it being the case that a monitored wireless unit in a local alarm system is considered to be able to send messages as long as a uniquely identifiable response signal is received therefrom.

4. Arrangement of units to form a monitoring system as per one or more of patent claims 1 to 3, said arrangement including at least one local alarm system that has at least one surveillance camera, said system for the transmission of alarms

comprising one or more monitoring stations, at least one of which has a device for traffic distribution, the whole being **characterised by** said traffic distribution device controlling the traffic load from surveillance cameras by: sending images and/or video streams to various receivers, the resolution and update frequency of the content corresponding to the available bandwidth and/or what the recipient requires of the images/video streams; using mathematical algorithms to compress said images and video streams (in order to take as little bandwidth as possible); and, determining to which receivers the images or video streams from surveillance cameras shall be sent.

5. Arrangement of units to form a monitoring system as per one or more of patent claims 1 to 4, said arrangement including at least one local alarm system and one or more monitoring stations, at least one of these latter having a device for remote configuration, the whole being **characterised by** said remote configuration device being able to update and/or install software in various units in a local alarm system, it being possible for such update to be effected using, for example, TCP/IP or other communication method.

6. Arrangement of units to form a monitoring system as per one or more of patent claims 1 to 5, said arrangement including at least one local alarm system and one or more monitoring stations, the whole being **characterised by** each and every one of the units in the local alarm system and in the monitoring stations having a unique piece of digital information that identifies the unit to which it belongs, there being, in addition, means of detecting if a unit with such a piece of identifying information is replaced.

7. Arrangement of units to form a monitoring system as per one or more of patent claims 1 to 6, said arrangement including at least one local alarm system and one or more monitoring stations, the whole being **characterised by** the possibility of giving the messages that are sent between local alarm systems and monitoring stations, and between the monitoring stations themselves, a uniquely identifiable piece of identification information composed of information about the message and its sender, this ensuring that if any message is changed in any way during its

transmission, the identification information for such a message does not "match" (i.e. have the correct profile for) the message.

8. Arrangement of units to form a monitoring system as per one or more of patent claims 1 to 7, said arrangement including at least one local alarm system, the whole being **characterised by** communication within local alarm systems being encrypted.

9. Arrangement of units to form a monitoring system as per one or more of patent claims 1 to 8, said arrangement including at least one local alarm system and one or more monitoring stations, the whole being **characterised by** the possibility of communication between a local alarm system and monitoring stations being encrypted and, similarly, the possibility that communication between different monitoring stations can be encrypted.

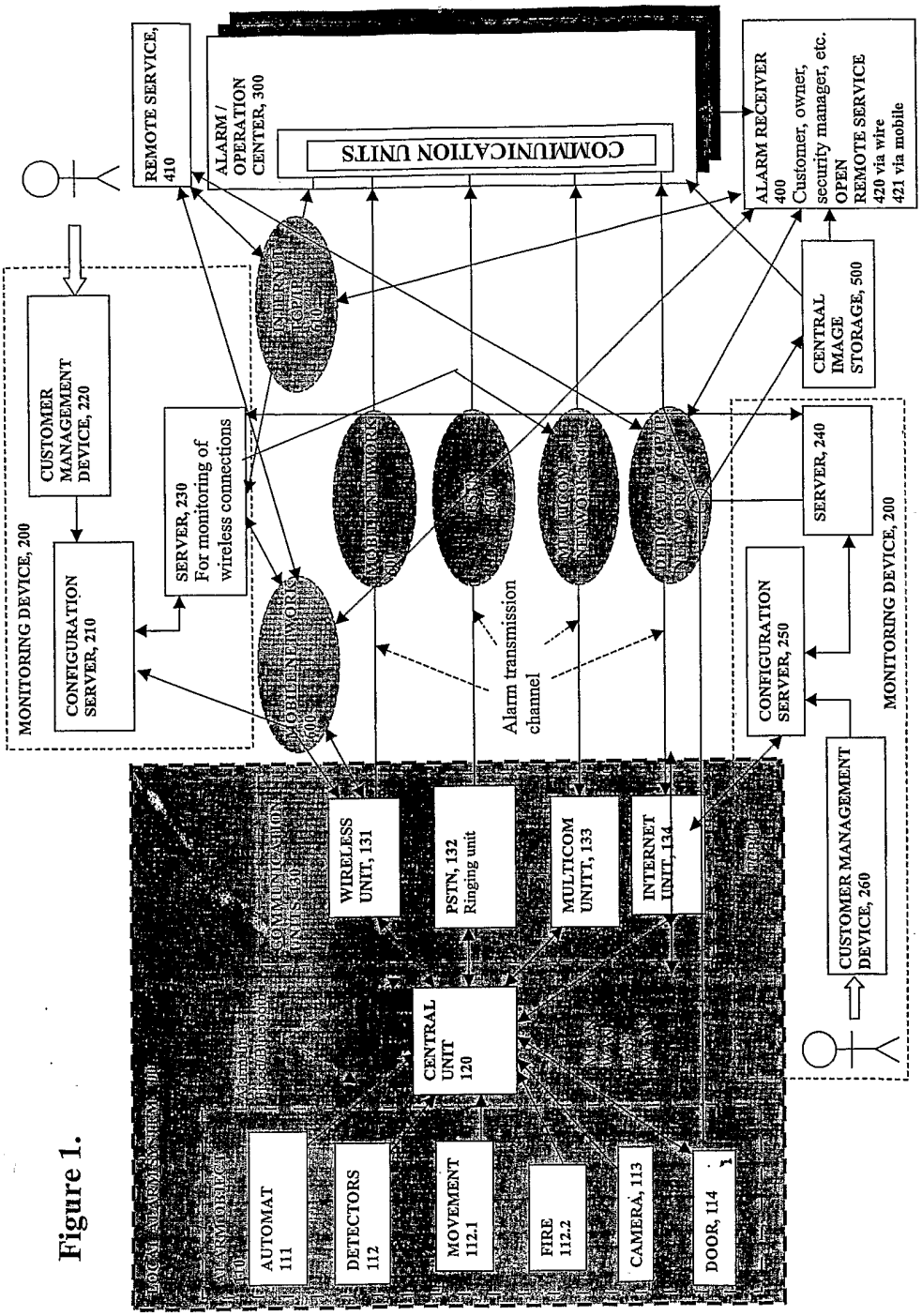

REMOTE SERVICE, 410
ALARM / OPERATION CENTER, 300
COMMUNICATION UNITS
ALARM RECEIVER 400
Customer, owner, security manager, etc.
OPEN REMOTE SERVICE
420 via wire
421 via mobile
CENTRAL IMAGE STORAGE, 500
CUSTOMER MANAGEMENT DEVICE, 220
SERVER, 230
For monitoring of wireless connections
MONITORING DEVICE, 200
CONFIGURATION SERVER, 210
INTERNET 430
MOBILE NETWORK 440
PSTN 450
MULTICOM NETWORK 460
DEDICATED NETWORK 470
Alarm transmission channel
SERVER, 240
CONFIGURATION SERVER, 250
MONITORING DEVICE, 200
CUSTOMER MANAGEMENT DEVICE, 260
COMMUNICATION UNITS 130
WIRELESS UNIT, 131
PSTN, 132 Ringing unit
MULTICOM UNITT, 133
INTERNET UNIT, 134
CENTRAL UNIT 120
AUTOMAT 111
DETECTORS 112
MOVEMENT 112.1
FIRE 112.2
CAMERA, 113
DOOR, 114

Figure 1.

Figure 2.

A FLOW DIAGRAM FOR AN ALARM IN FIGURE 1

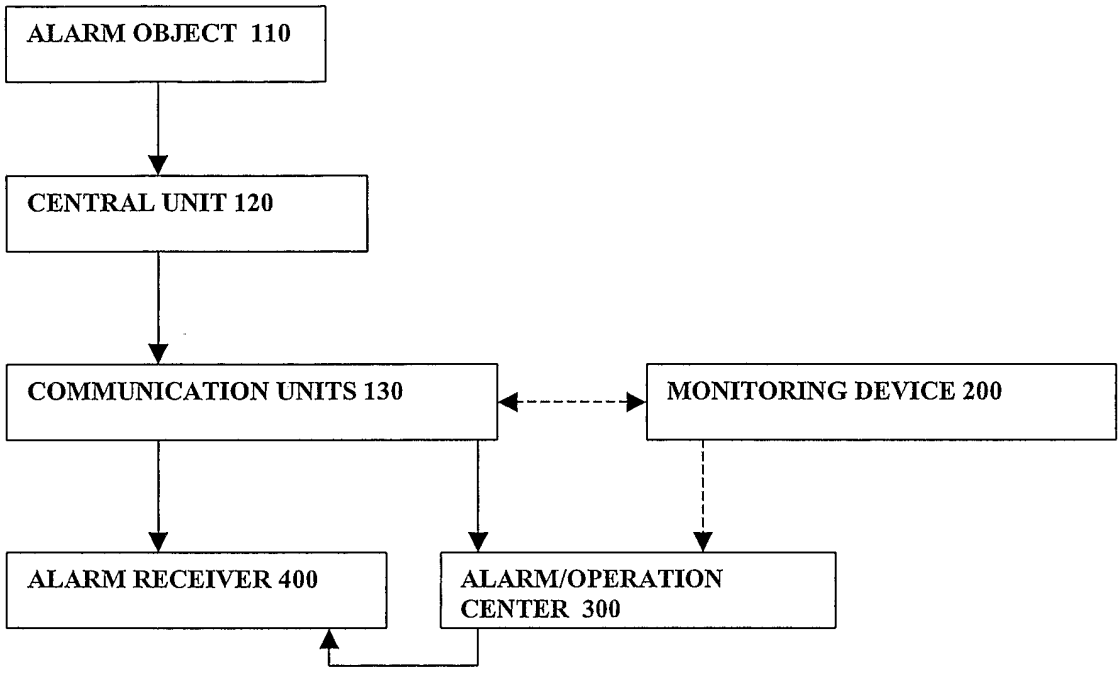


Figure 3

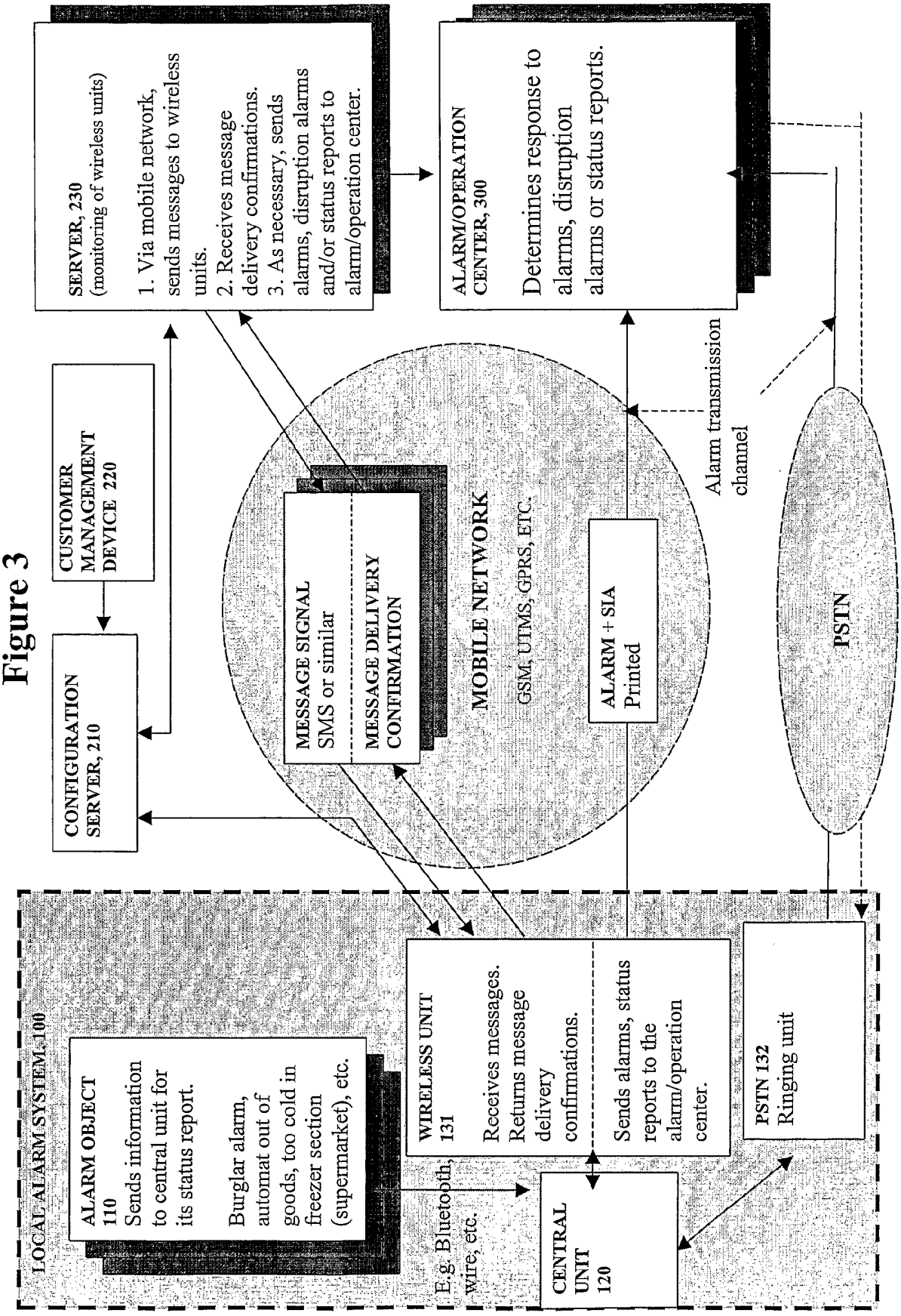

LOCAL ALARM SYSTEM, 100
ALARM OBJECT 110
Sends information to central unit for its status report.
Burglar alarm, automat out of goods, too cold in freezer section (supermarket), etc.
E.g. Bluetooth, wire, etc.
WIRELESS UNIT 131
Receives messages. Returns message delivery confirmations.
Sends alarms, status reports to the alarm/operation center.
CENTRAL UNIT 120
PSTN 132
Ringing unit
CONFIGURATION SERVER, 210
CUSTOMER MANAGEMENT DEVICE 220
SERVER, 230
(monitoring of wireless units)
1. Via mobile network, sends messages to wireless units.
2. Receives message delivery confirmations.
3. As necessary, sends alarms, disruption alarms and/or status reports to alarm/operation center.
ALARM/OPERATION CENTER, 300
Determines response to alarms, disruption alarms or status reports.
MESSAGE SIGNAL
SMS or similar
MESSAGE DELIVERY CONFIRMATION
MOBILE NETWORK
GSM, UTMS, GPRS, ETC.
ALARM + SIA
Printed
Alarm transmission channel
PSTN

Figure 4.

FLOW DIAGRAM FOR AN ALARM IN THAT PART OF THE MONITORING SYSTEM SHOWN IN FIGURE 3

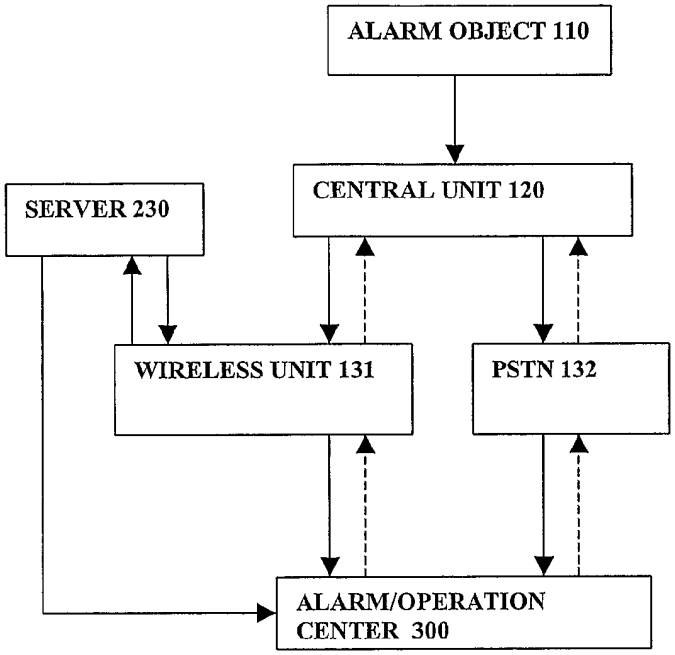


Figure 5.

A FLOW DIAGRAM FOR THE CONFIGURATION AND MONITORING OF A WIRELESS UNIT

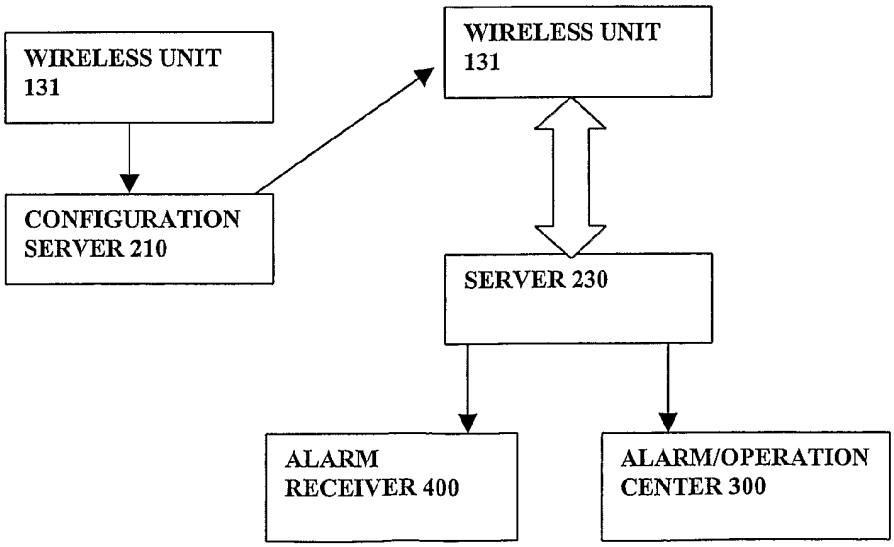


Figure 6.

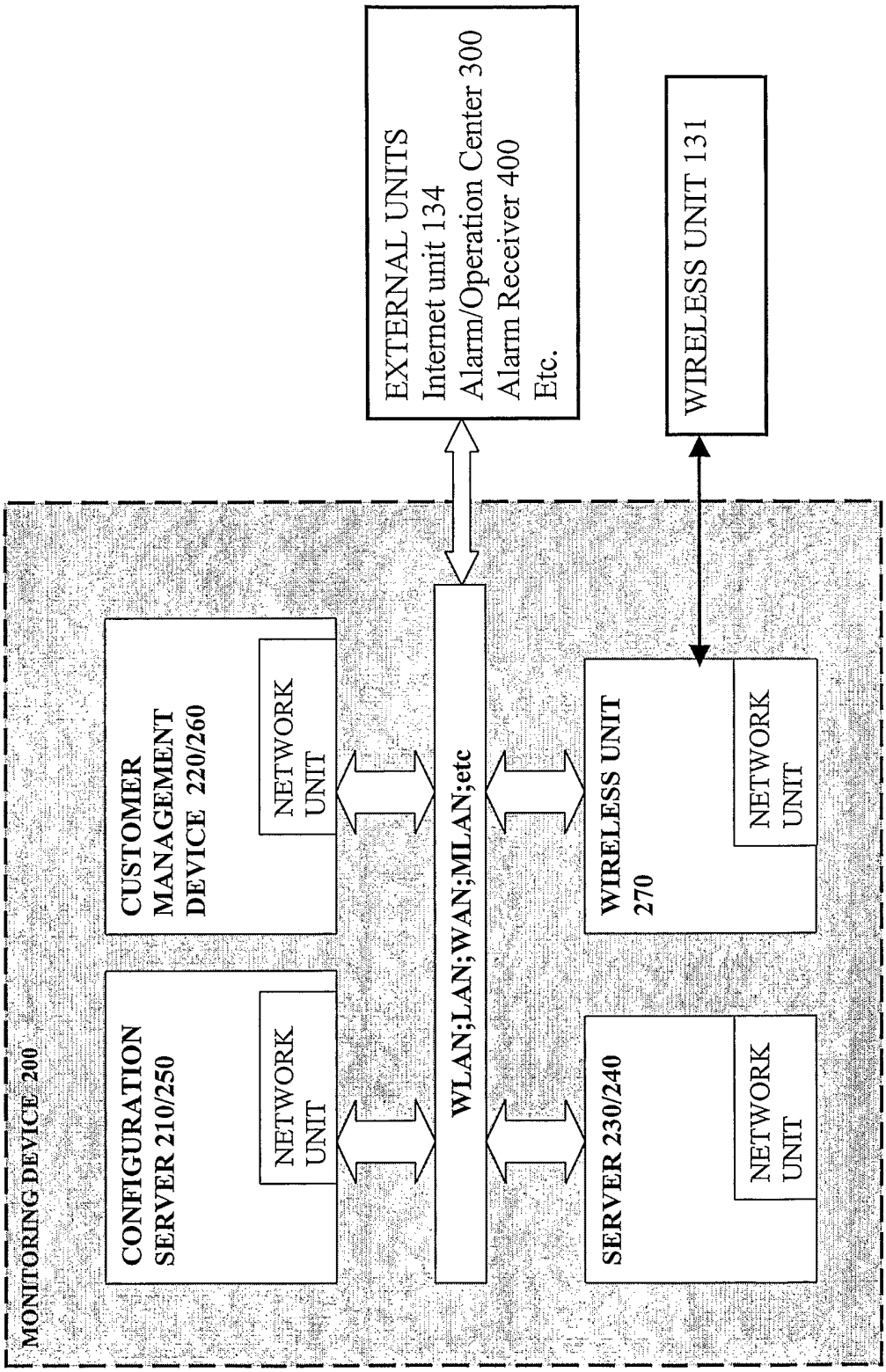
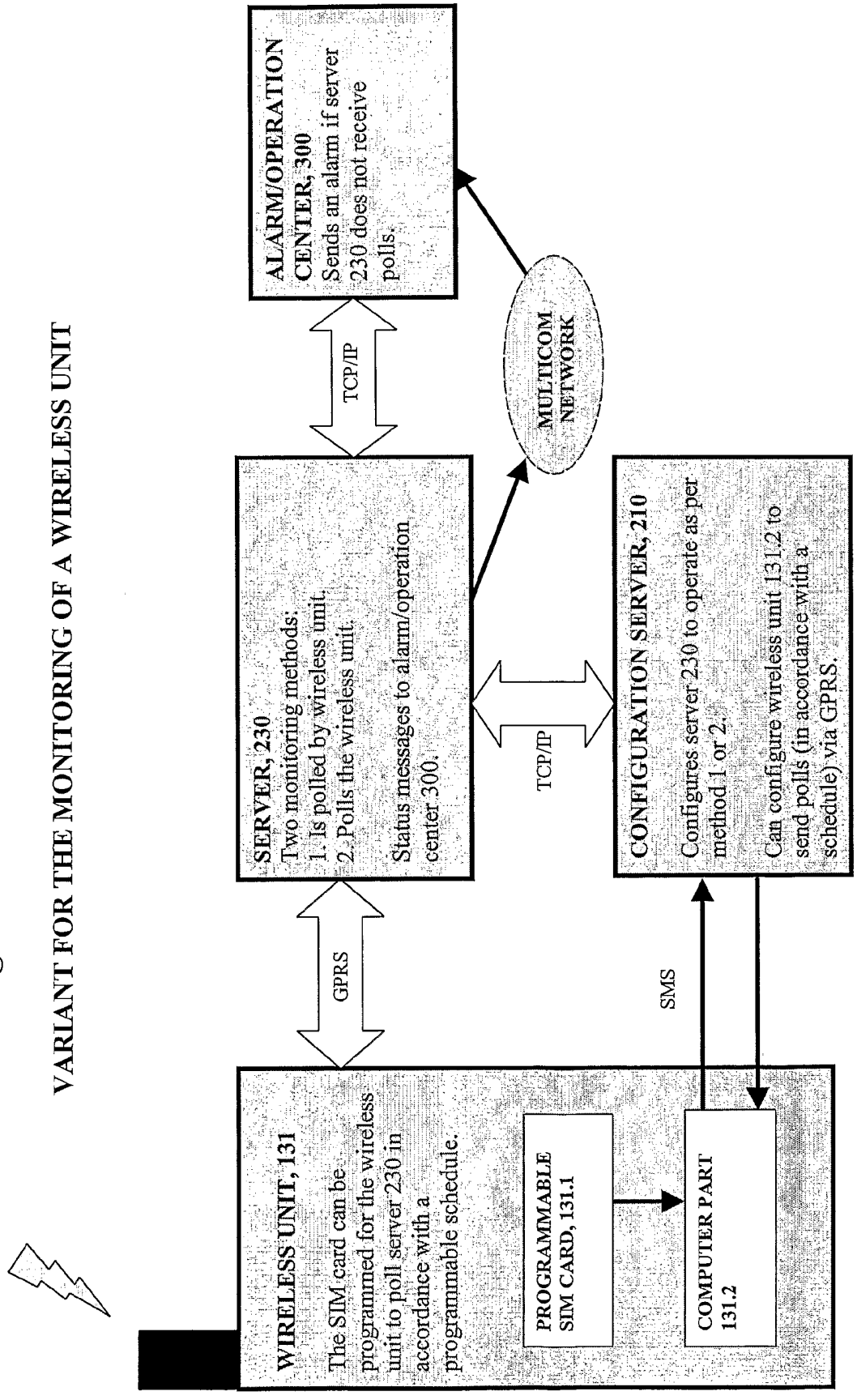

EXTERNAL UNITS
Internet unit 134
Alarm/Operation Center 300
Alarm Receiver 400
Etc.
WIRELESS UNIT 131
MONITORING DEVICE 200
CONFIGURATION SERVER 210/250
NETWORK UNIT
CUSTOMER MANAGEMENT DEVICE 220/260
NETWORK UNIT
WLAN;LAN;WAN;MLAN;etc
SERVER 230/240
NETWORK UNIT
WIRELESS UNIT 270
NETWORK UNIT

Figure 7

VARIANT FOR THE MONITORING OF A WIRELESS UNIT



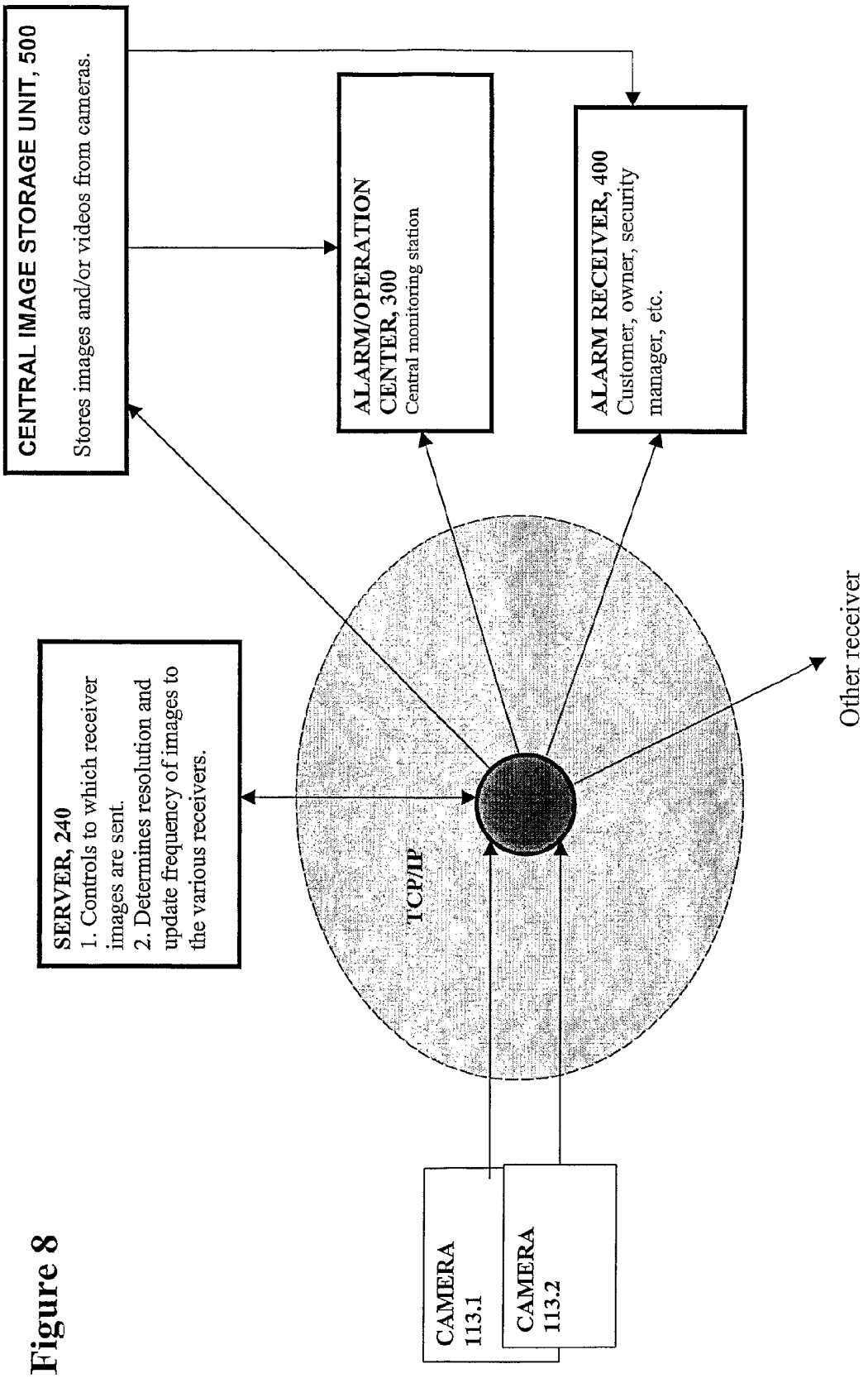

CENTRAL IMAGE STORAGE UNIT, 500
Stores images and/or videos from cameras.
ALARM/OPERATION CENTER, 300
Central monitoring station
ALARM RECEIVER, 400
Customer, owner, security manager, etc.
SERVER, 240
1. Controls to which receiver images are sent.
2. Determines resolution and update frequency of images to the various receivers.
TCP/IP
Other receiver
CAMERA 113.1
CAMERA 113.2

Figure 8

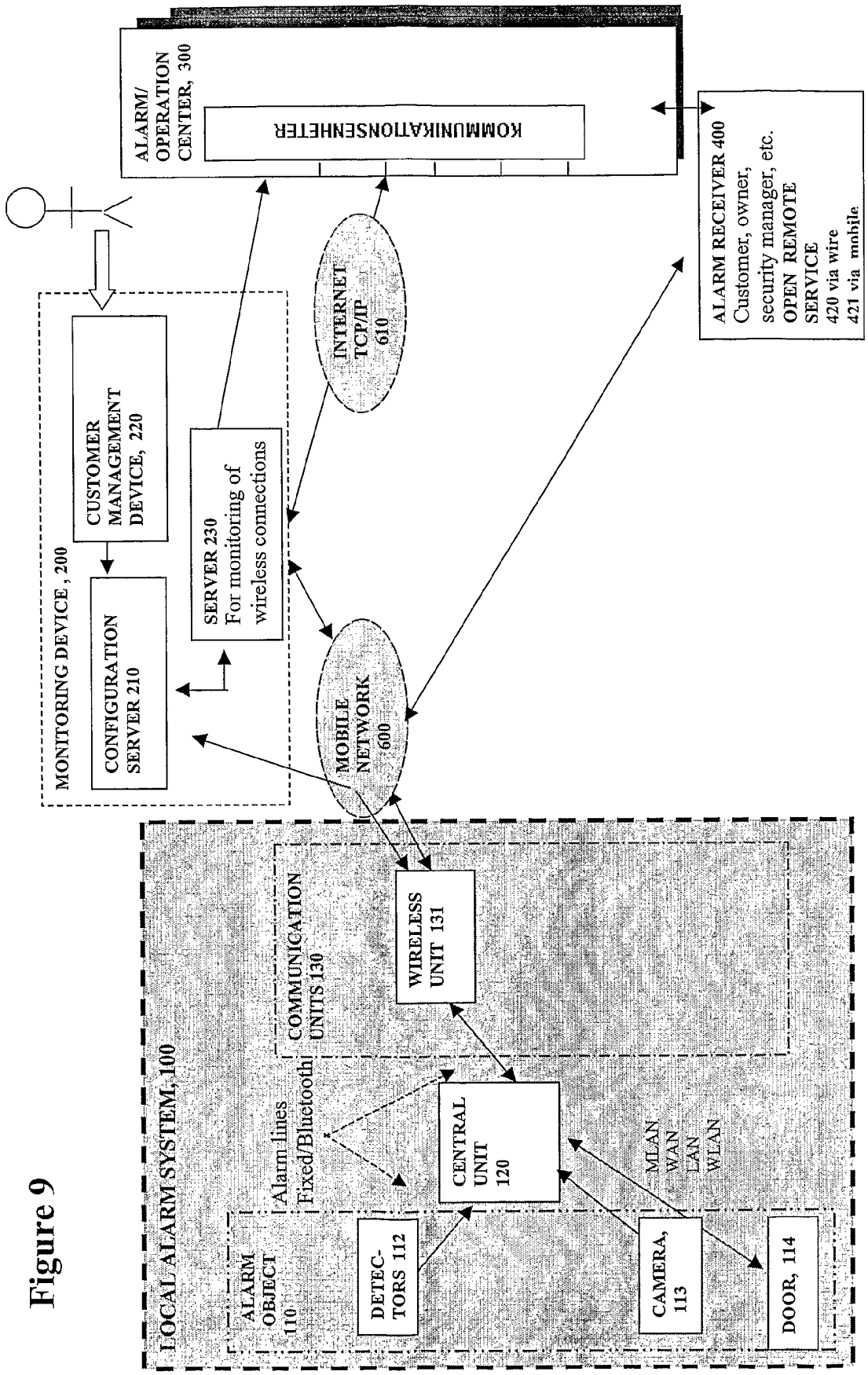

ALARM/
OPERATION
CENTER, 300
KOMMUNIKATIONSENHETER
ALARM RECEIVER 400
Customer, owner,
security manager, etc.
OPEN REMOTE
SERVICE
420 via wire
421 via mobile
INTERNET
TCP/IP
610
MONTORING DEVICE , 200
CUSTOMER
MANAGEMENT
DEVICE, 220
CONFIGURATION
SERVER 210
SERVER 230
For monitoring of
wireless connections
MOBILE
NETWORK
600
LOCAL ALARM SYSTEM, 100
ALARM
OBJECT
110
COMMUNICATION
UNITS 130
Alarm lines
Fixed/Bluetooth
WIRELESS
UNIT 131
CENTRAL
UNIT
120
DETEC-
TORS 112
CAMERA,
113
DOOR, 114
MLAN
WAN
LAN
WLAN

Figure 9

Figure 10

Flow Diagram Home

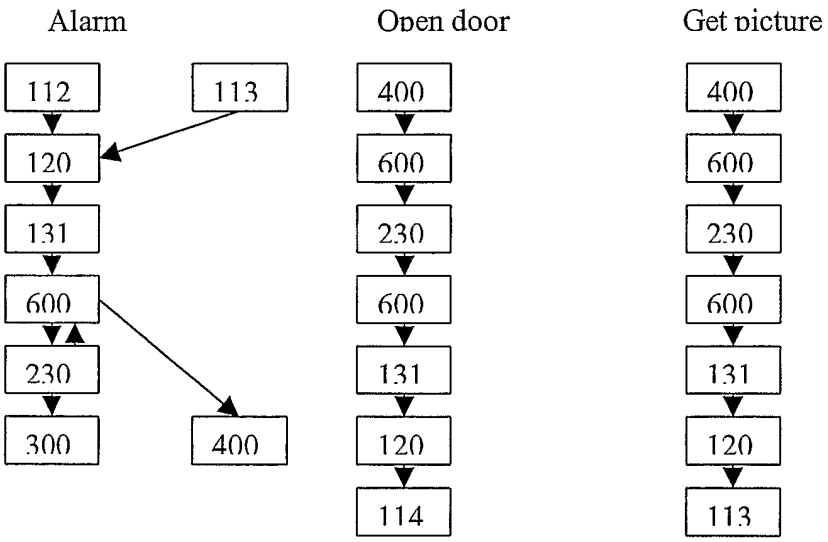
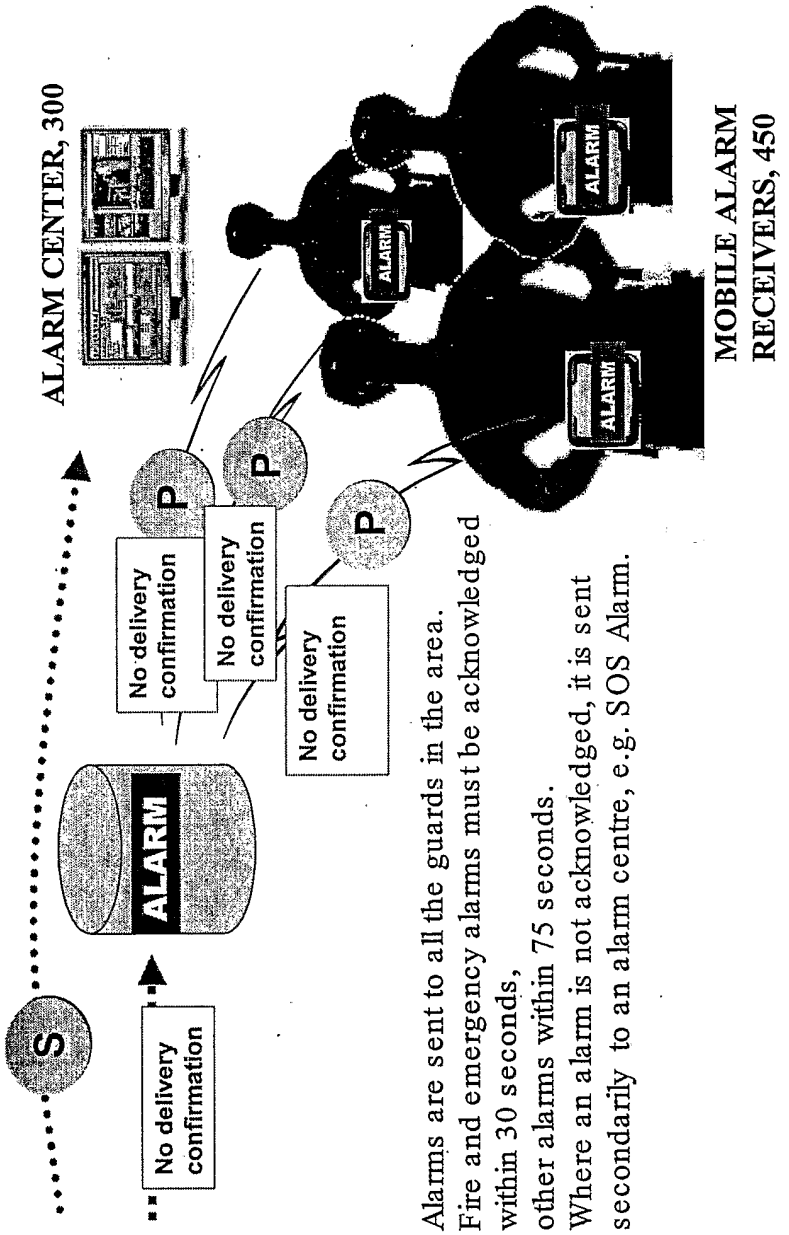
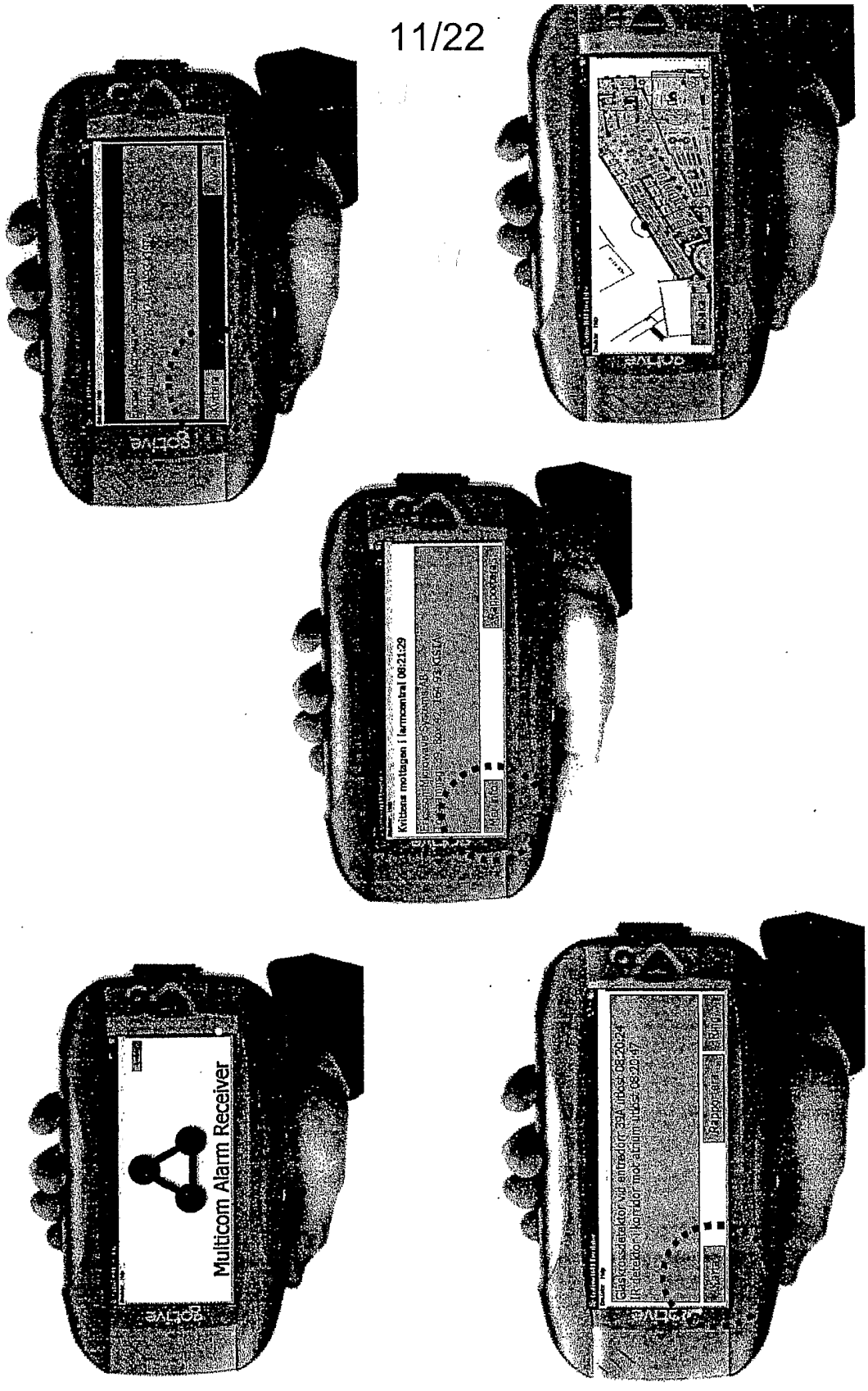
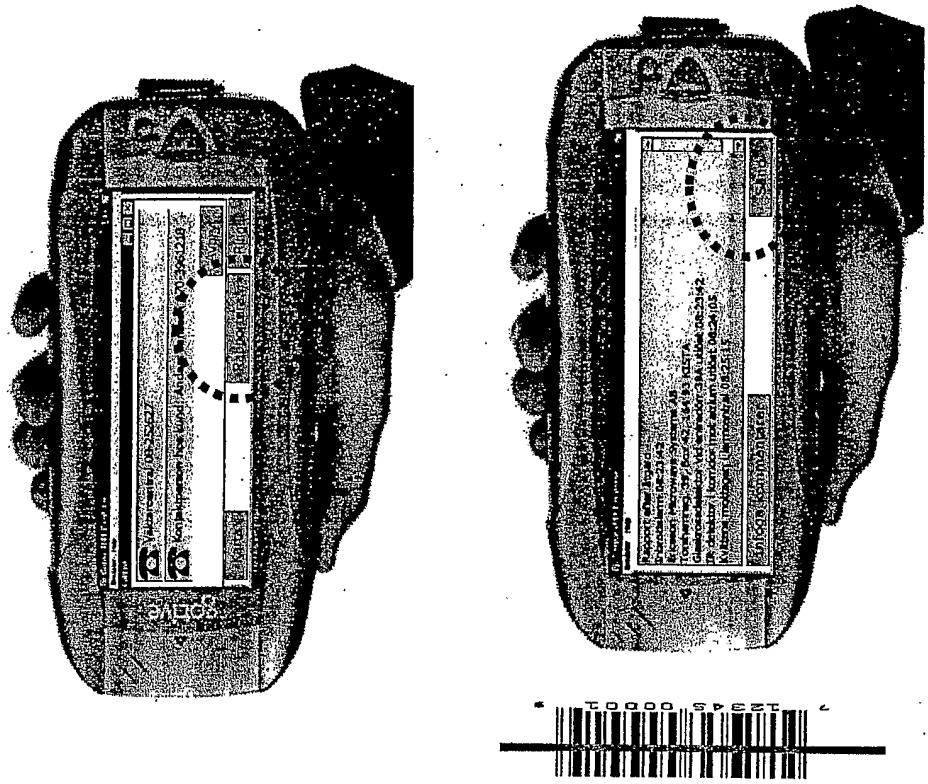


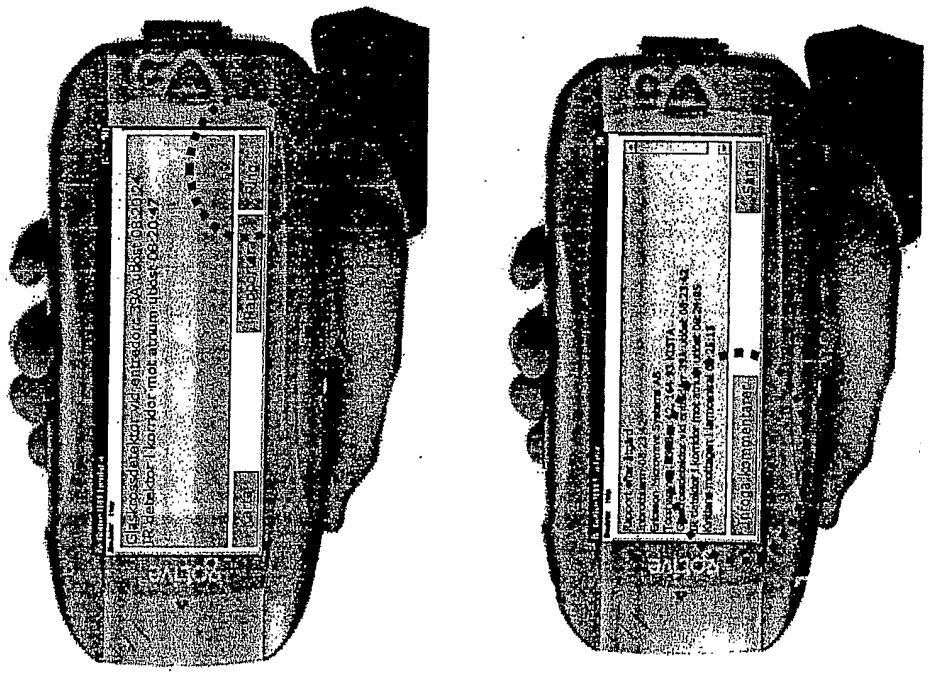
Figure 11


Principal events in an alarm
ALARM CENTER, 300
MOBILE ALARM RECEIVERS, 450
S
No delivery confirmation
ALARM
P
No delivery confirmation
P
No delivery confirmation
P
No delivery confirmation
ALARM
ALARM
ALARM
Alarms are sent to all the guards in the area.
Fire and emergency alarms must be acknowledged within 30 seconds,
other alarms within 75 seconds.
Where an alarm is not acknowledged, it is sent secondarily to an alarm centre, e.g. SOS Alarm.



Figur 12





Figur 13

Figure 14

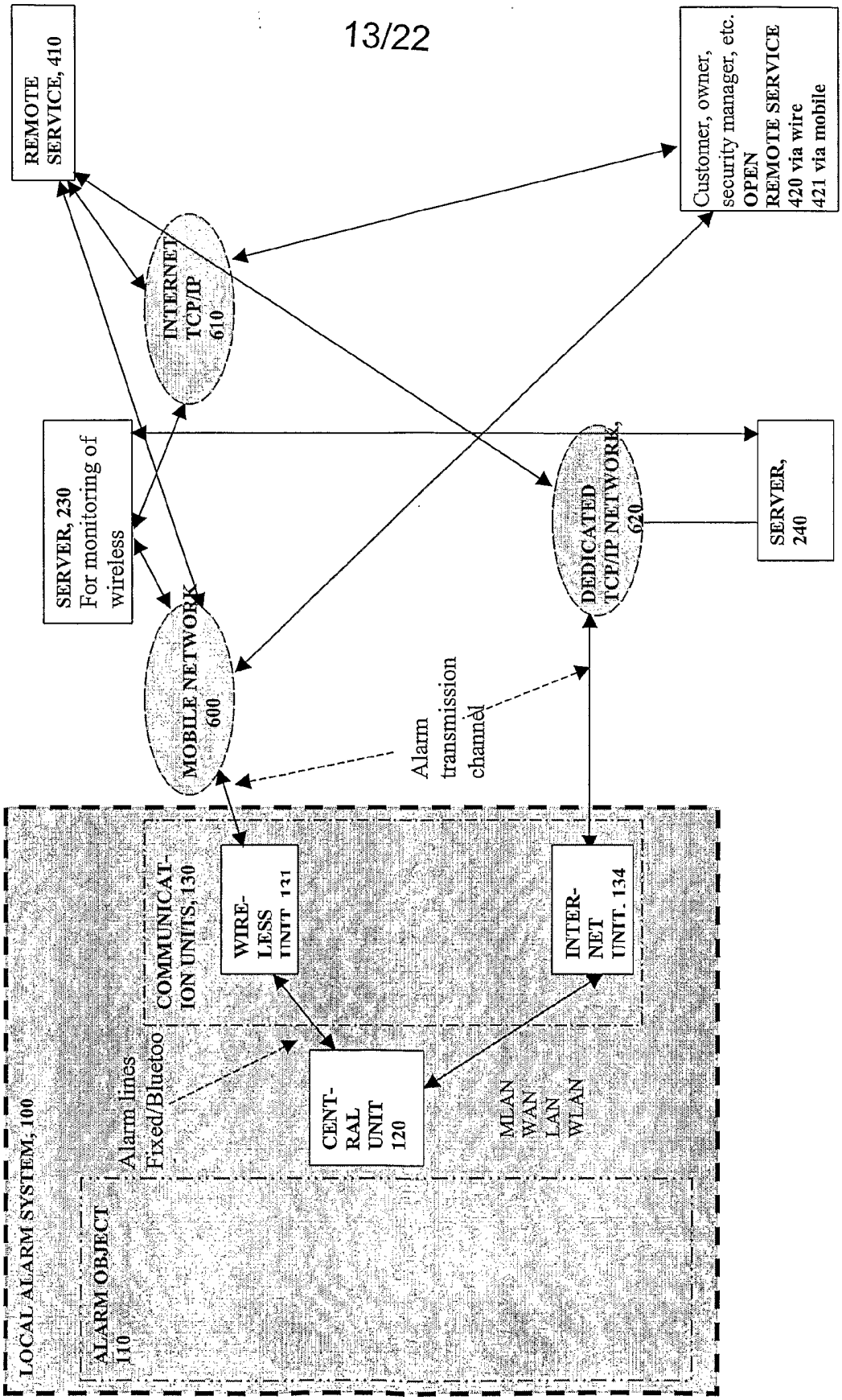
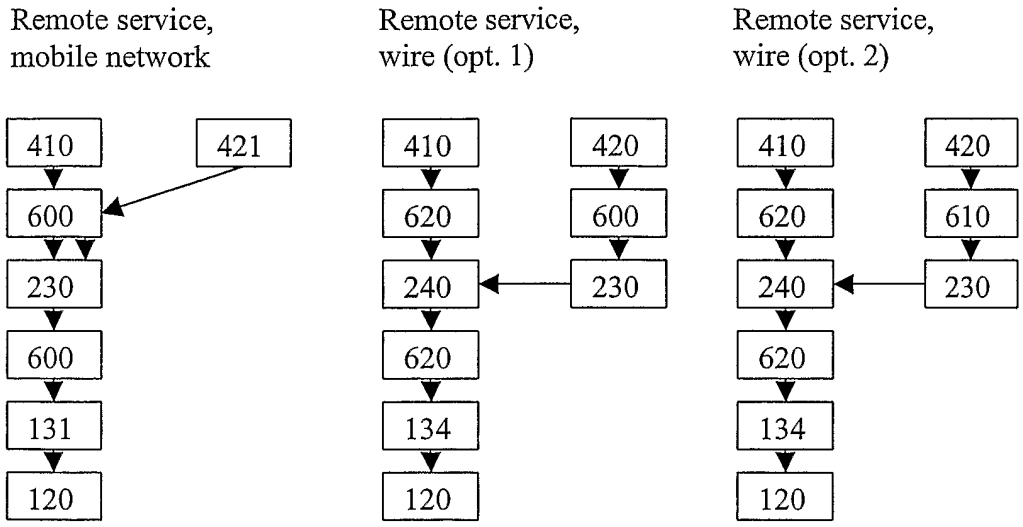

REMOTE SERVICE, 410
INTERNET TCP/IP 610
Customer, owner, security manager, etc. OPEN REMOTE SERVICE 420 via wire 421 via mobile
SERVER, 230 For monitoring of wireless
MOBILE NETWORK 600
DEDICATED TCP/IP NETWORK, 620
SERVER, 240
Alarm transmission channel
LOCAL ALARM SYSTEM, 100
ALARM OBJECT 110
Alarm lines
Fixed/Bluetoo
COMMUNICAT-ION UNITS, 130
WIRE-LESS UNIT 131
INTER-NET UNIT 134
CENT-RAL UNIT 120
MLAN
WAN
LAN
WLAN

Figure 15

Service flow diagram



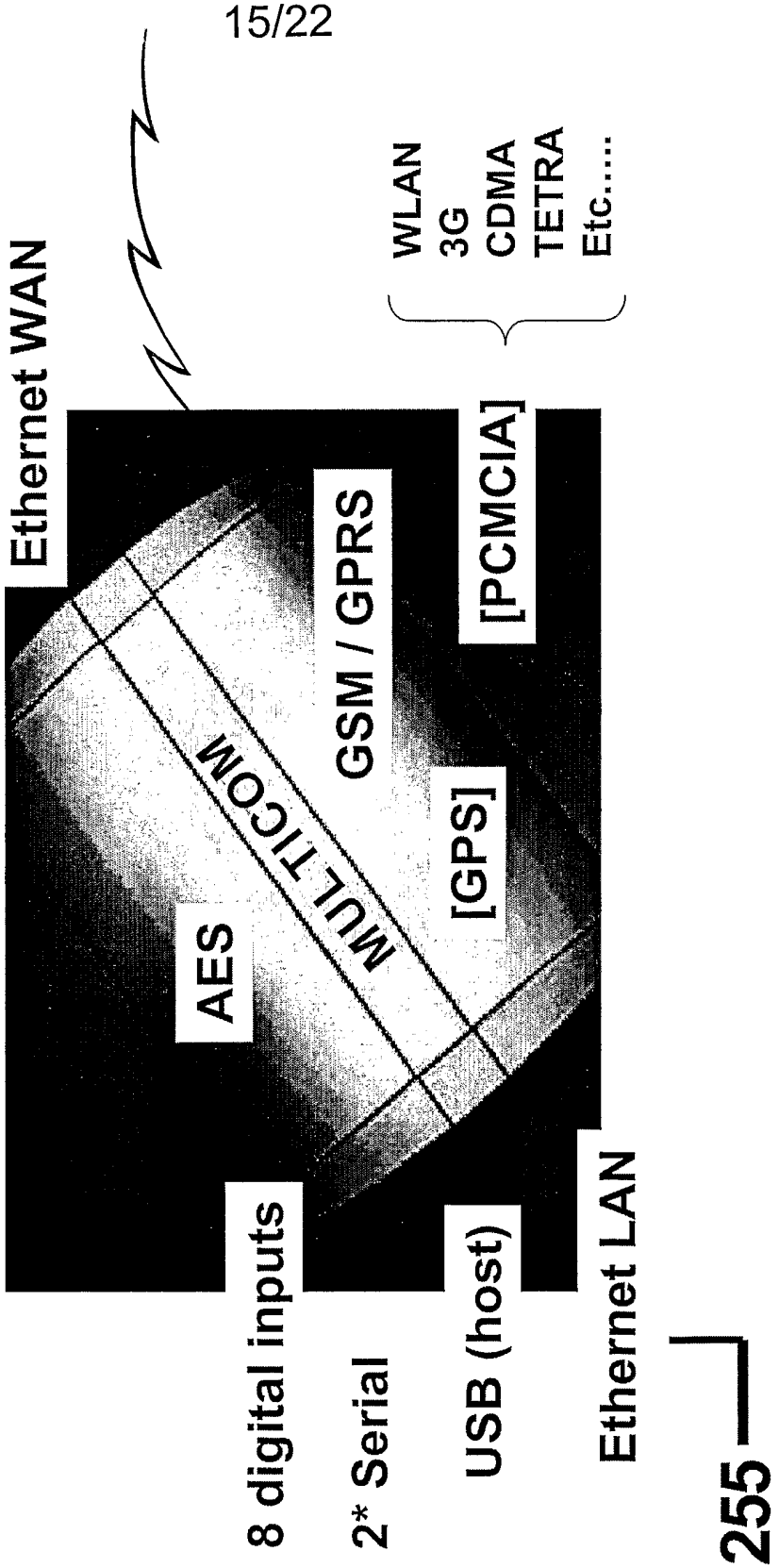

FIXED & WIRELESS IP TERMINAL
Ethernet WAN
WLAN
3G
CDMA
TETRA
Etc.....
AES
MULTICOM
GSM / GPRS
[GPS]
[PCMCIA]
8 digital inputs
2* Serial
USB (host)
Ethernet LAN
255

Figure 16

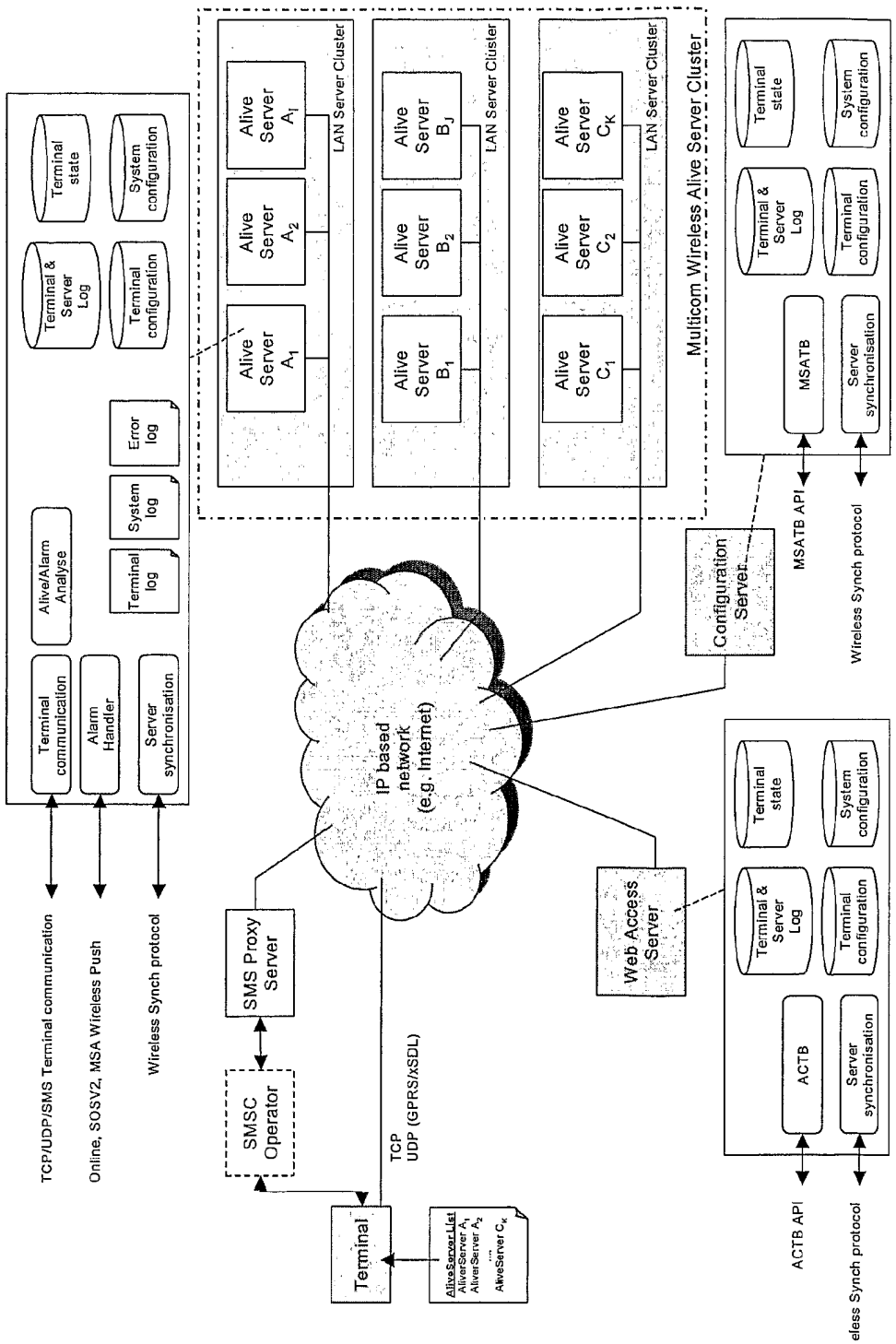

Multicom Wireless Alive Server Cluster
LAN Server Cluster
Alive Server A1
Alive Server A2
Alive Server AI
Alive Server B1
Alive Server B2
Alive Server BJ
Alive Server C1
Alive Server C2
Alive Server CK
Terminal state
System configuration
Terminal & Server Log
Terminal configuration
Error log
System log
Terminal log
Alive/Alarm Analyse
Terminal communication
Alarm Handler
Server synchronisation
MSATB
Configuration Server
MSATB API
Wireless Synch protocol
IP based network (e.g. Internet)
SMS Proxy Server
SMSC Operator
Web Access Server
ACTB
ACTB API
TCP
UDP (GPRS/xSDL)
Terminal
AliveServer List
AliverServer A1
AliverServer A2
AliveServer CK
TCP/UDP/SMS Terminal communication
Online, SOSV2, MSA Wireless Push
Wireless Synch protocol

Figure 17

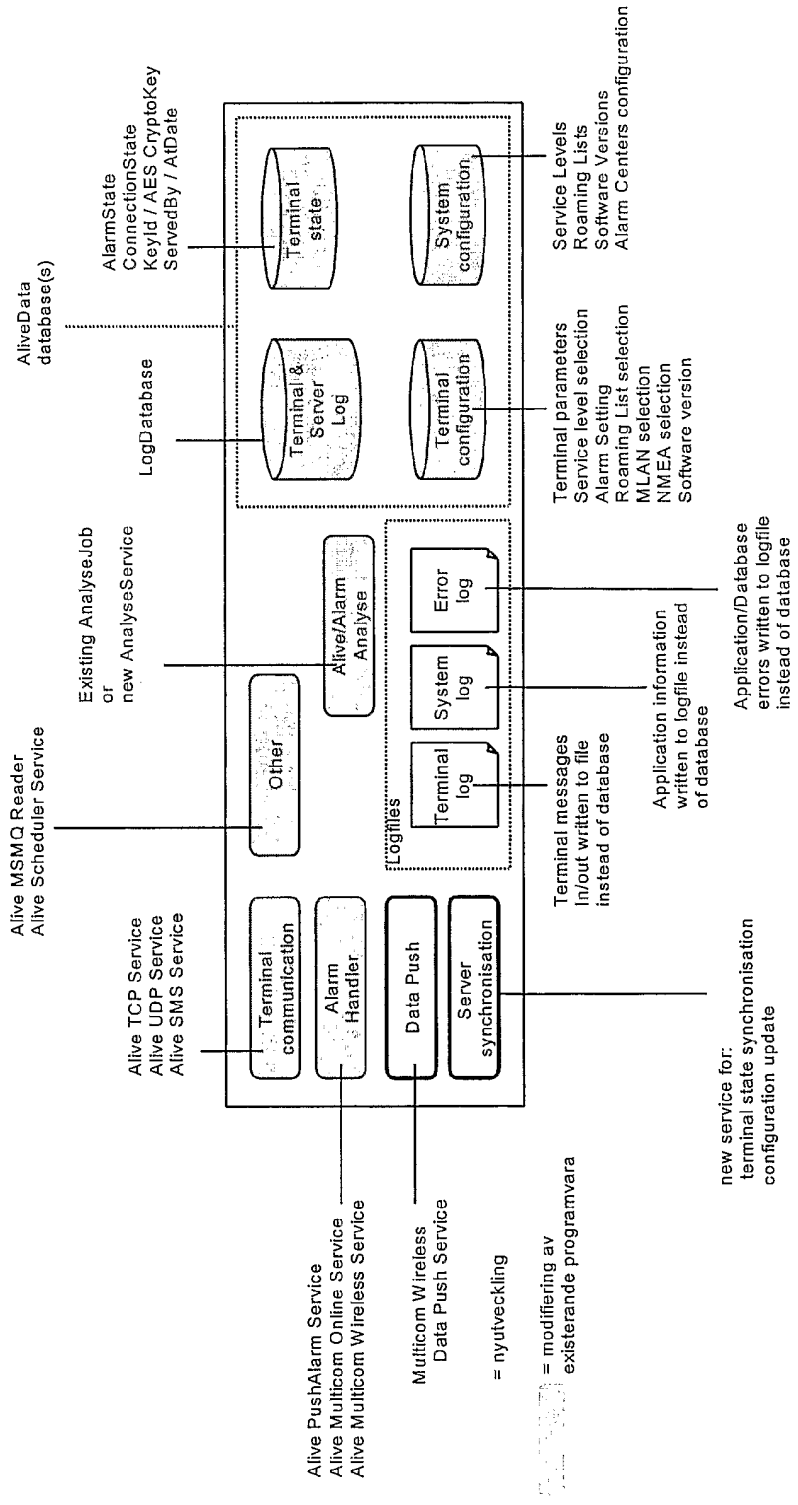

AliveData database(s)
AlarmState
ConnectionState
KeyId / AES CryptoKey
ServedBy / AtDate
Terminal state
System configuration
Service Levels
Roaming Lists
Software Versions
Alarm Centers configuration
LogDatabase
Terminal & Server Log
Terminal configuration
Terminal parameters
Service level selection
Alarm Setting
Roaming List selection
MLAN selection
NMEA selection
Software version
Existing AnalyseJob or new AnalyseService
Alive/Alarm Analyse
Error log
System log
Terminal log
Logfiles
Application/Database errors written to logfile instead of database
Application information written to logfile instead of database
Terminal messages In/out written to file instead of database
Alive MSMQ Reader
Alive Scheduler Service
Other
Alive TCP Service
Alive UDP Service
Alive SMS Service
Terminal communication
Alarm Handler
Data Push
Server synchronisation
new service for:
terminal state synchronisation
configuration update
Alive PushAlarm Service
Alive Multicom Online Service
Alive Multicom Wireless Service
Multicom Wireless Data Push Service
= nyutveckling
= modifiering av existerande programvara

Figure 18

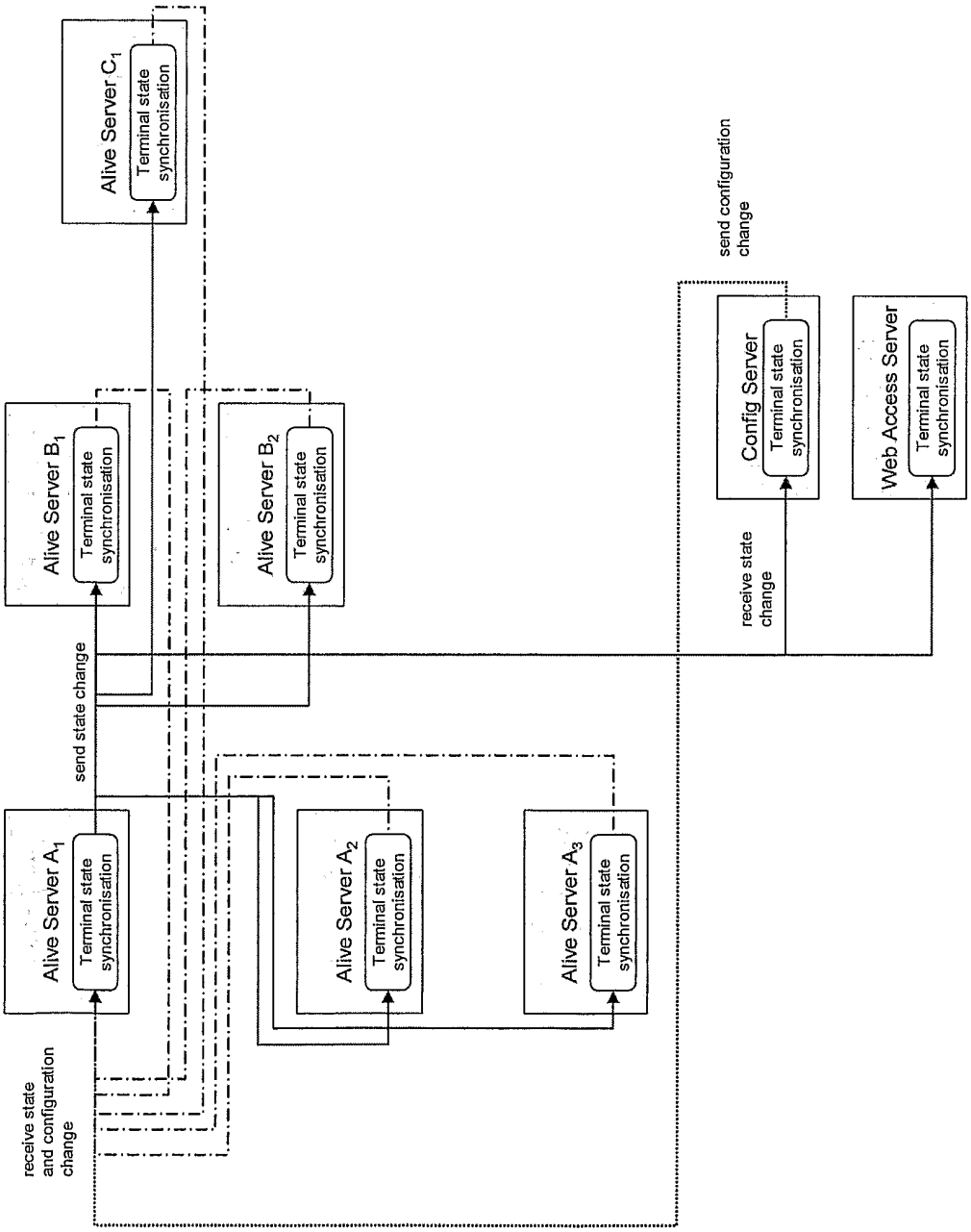
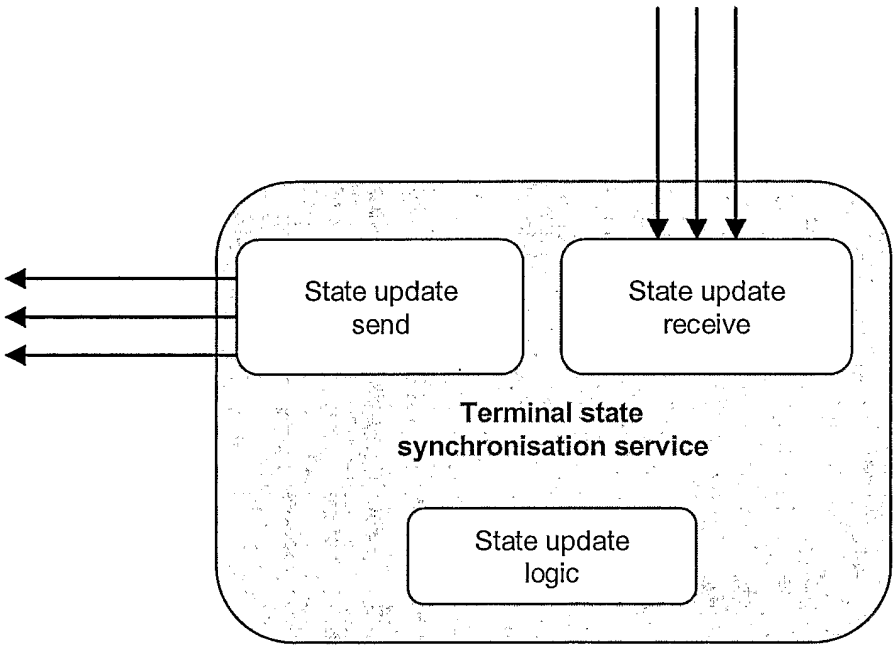

Alive Server C_1
Terminal state synchronisation
Alive Server B_1
Terminal state synchronisation
Alive Server B_2
Terminal state synchronisation
Config Server
Terminal state synchronisation
Web Access Server
Terminal state synchronisation
Alive Server A_1
Terminal state synchronisation
Alive Server A_2
Terminal state synchronisation
Alive Server A_3
Terminal state synchronisation
send configuration change
receive state change
send state change
receive state and configuration change

Figure 19

Figure 20


State update send
State update receive
Terminal state synchronisation service
State update logic

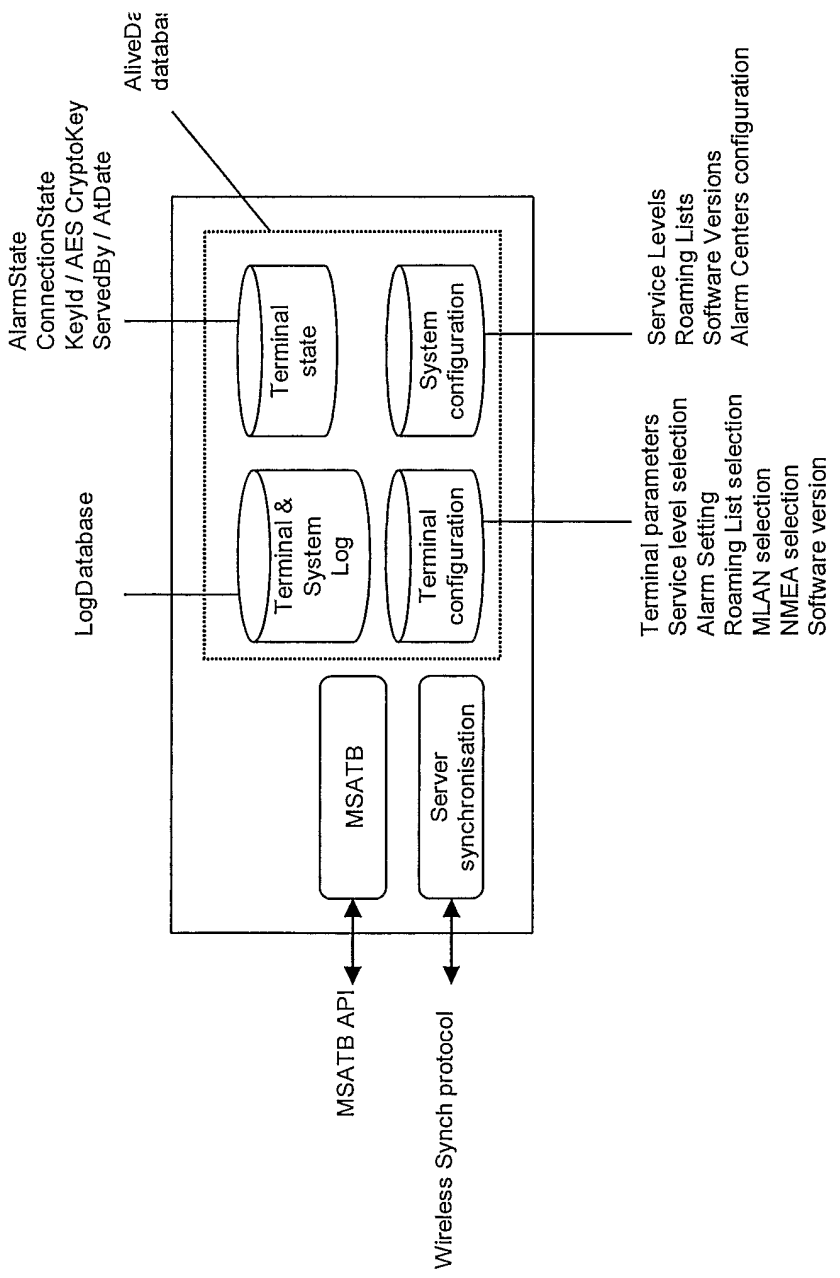

AliveDa
databa
AlarmState
ConnectionState
KeyId / AES CryptoKey
ServedBy / AtDate
LogDatabase
Terminal state
System configuration
Terminal & System Log
Terminal configuration
MSATB
Server synchronisation
Service Levels
Roaming Lists
Software Versions
Alarm Centers configuration
Terminal parameters
Service level selection
Alarm Setting
Roaming List selection
MLAN selection
NMEA selection
Software version
MSATB API
Wireless Synch protocol

Figure 21

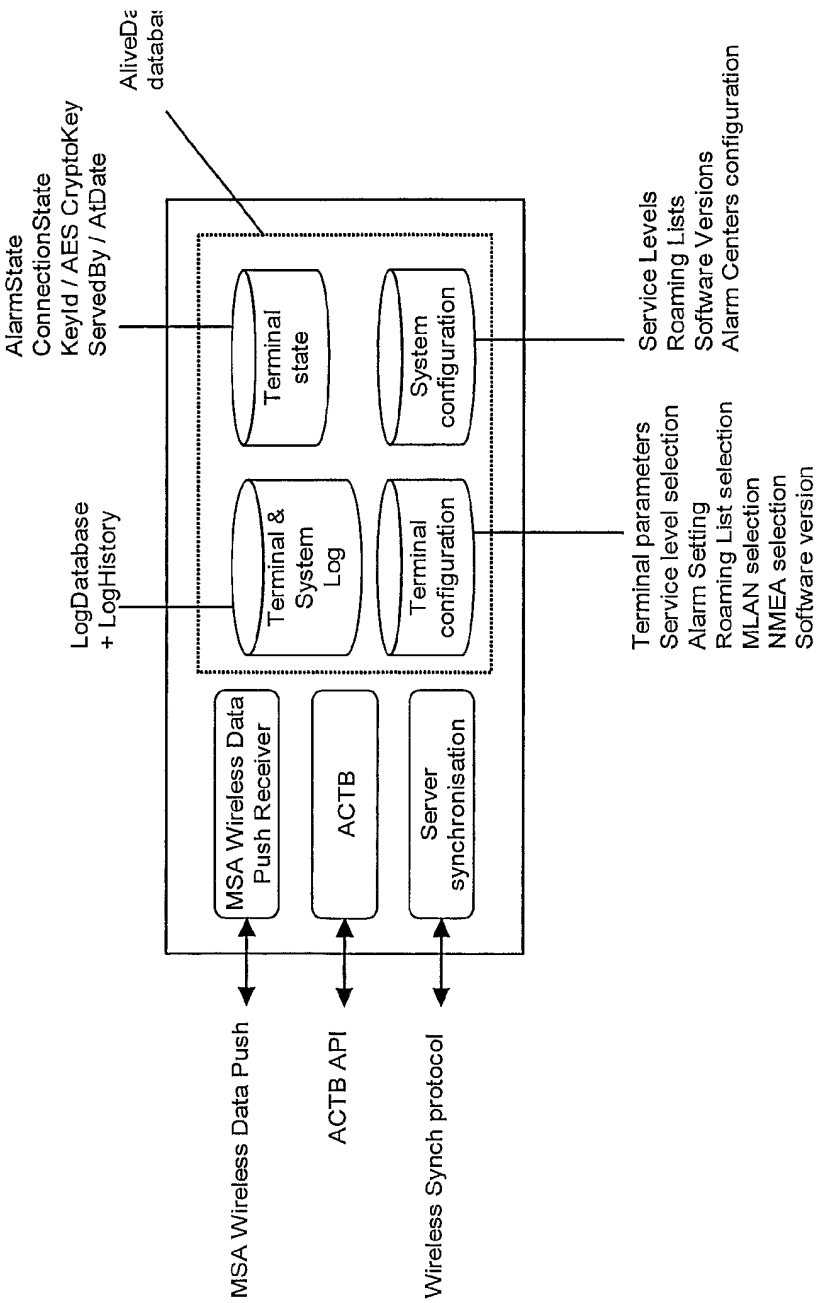

AliveDa
databa
AlarmState
ConnectionState
KeyId / AES CryptoKey
ServedBy / AtDate
LogDatabase
+ LogHistory
Terminal state
System configuration
Terminal & System Log
Terminal configuration
MSA Wireless Data Push Receiver
ACTB
Server synchronisation
Service Levels
Roaming Lists
Software Versions
Alarm Centers configuration
Terminal parameters
Service level selection
Alarm Setting
Roaming List selection
MLAN selection
NMEA selection
Software version
MSA Wireless Data Push
ACTB API
Wireless Synch protocol

Figure 22

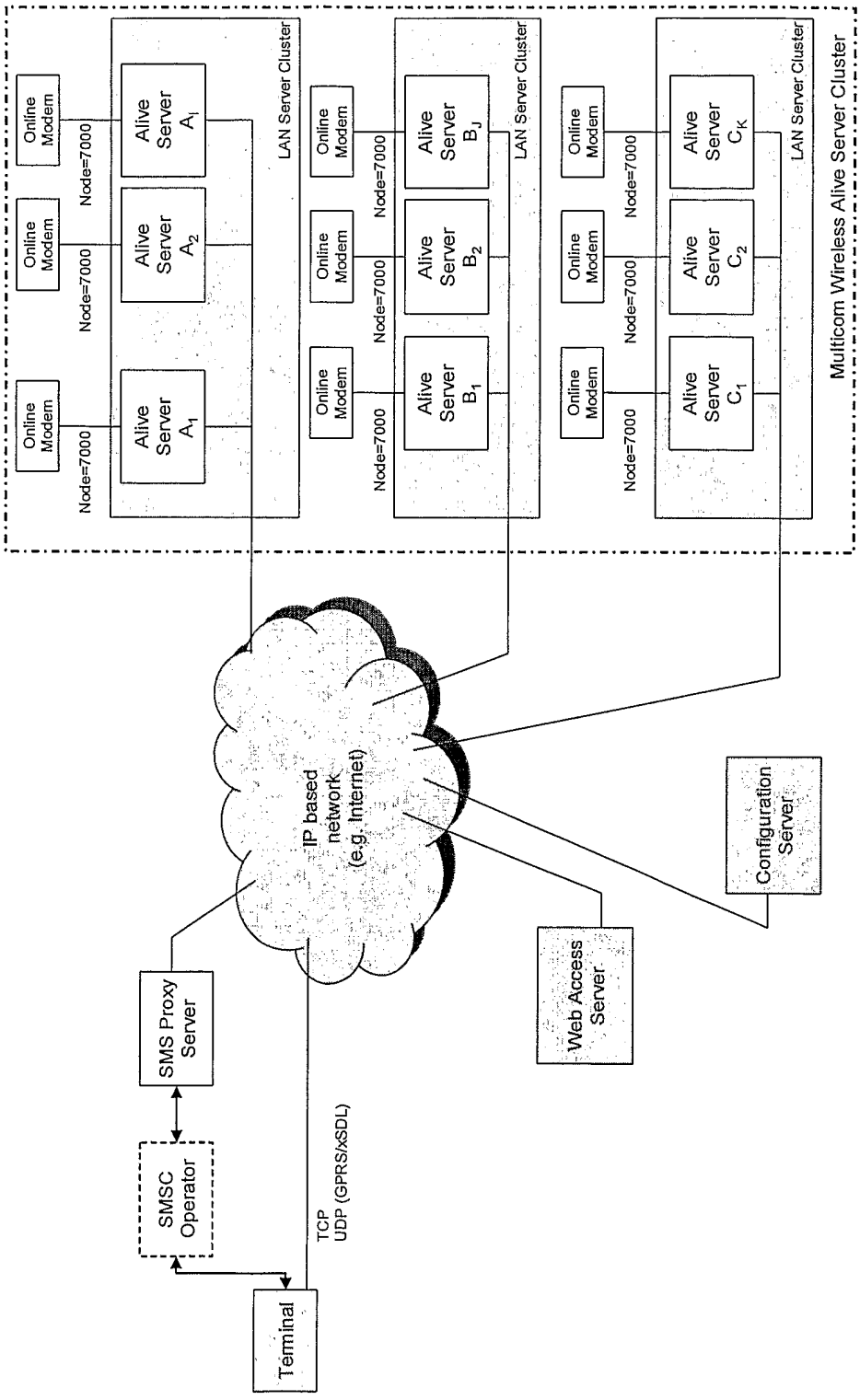

Multicom Wireless Alive Server Cluster
LAN Server Cluster
Alive Server A_1
Alive Server A_2
Alive Server A_I
Alive Server B_1
Alive Server B_2
Alive Server B_J
Alive Server C_1
Alive Server C_2
Alive Server C_K
Online Modem
Node=7000
IP based network (e.g. Internet)
Configuration Server
Web Access Server
SMS Proxy Server
SMSC Operator
Terminal
TCP
UDP (GPRS/xSDL)

Figure 23