

Call Button System Software

Technical Description

ADE Vertriebs-GmbH
D – 53288 Nörvenich
Gewerbepark 5

Autor: Detlef Reil, 01/2003
dr@itdex.de

1 Call Button System Software

Die Call Button System Software bildet die Schnittstelle zwischen den Call Buttons (Hardware) und der Applikation. Sie führt die direkte Kommunikation mit den Call Buttons durch und erfüllt folgende Aufgaben:

- Sichere Abwicklung der Kommunikation
- Zentrale Ansprechstelle für die Applikation
- Überwachung der Funktion der Call Buttons
- Administration der Call Buttons:
 - o Konfiguration
 - o Automatisches Firmware-Update
 - o Warnung bei Batterie leer
 - o Monitoring / Logging der Call Buttons und der Kommunikation
- Minimierung der Einschaltdauer der Call Buttons, um eine hohe Betriebszeit der Batterien zu erreichen
- Minimierung der Netzwerkbelastung

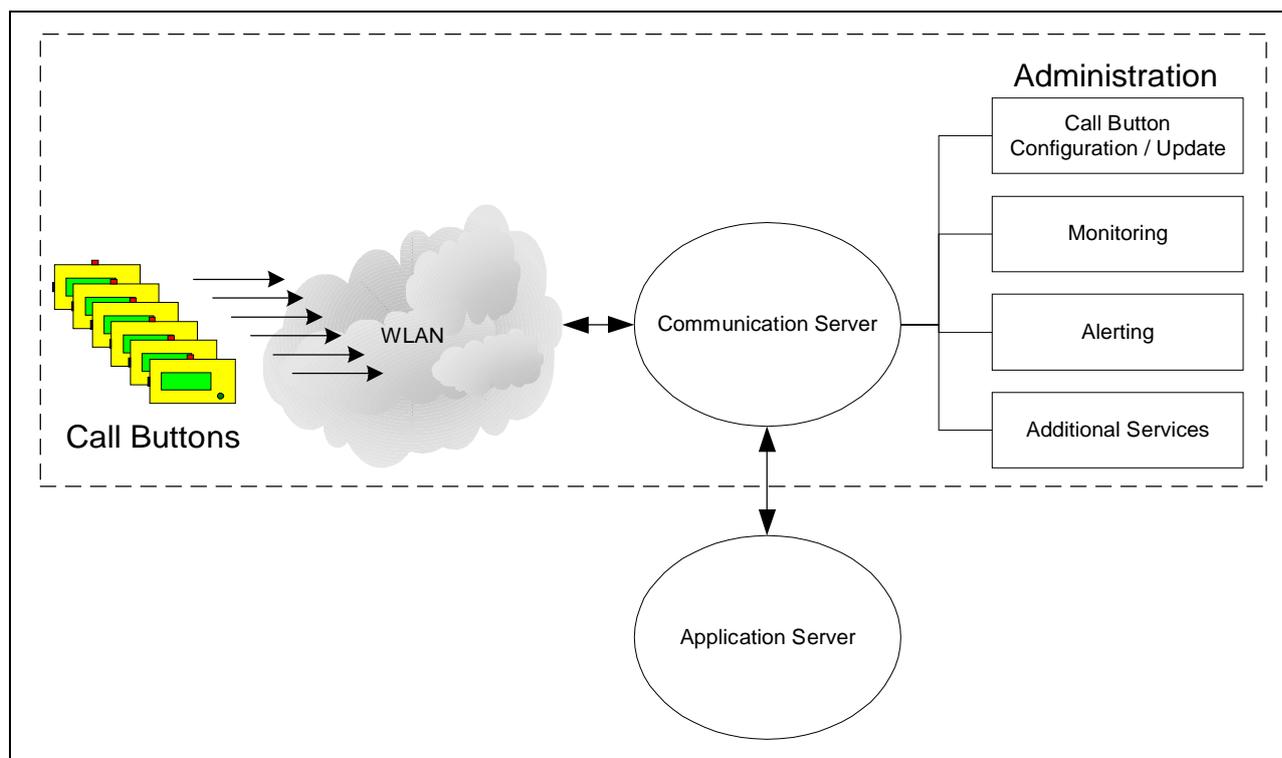


Abb. 1.1 – Call Button System

2 Der tronic Call Button

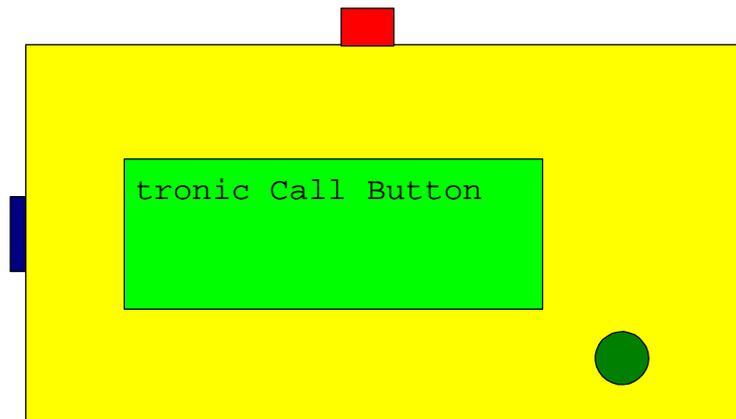


Abb. 2.1 – Ansicht Call Button

2.1 Funktionsweise

Durch ein Ereignis wird der tronic Call Button eingeschaltet, baut eine Verbindung zum Server auf und teilt das Ereignis mit.

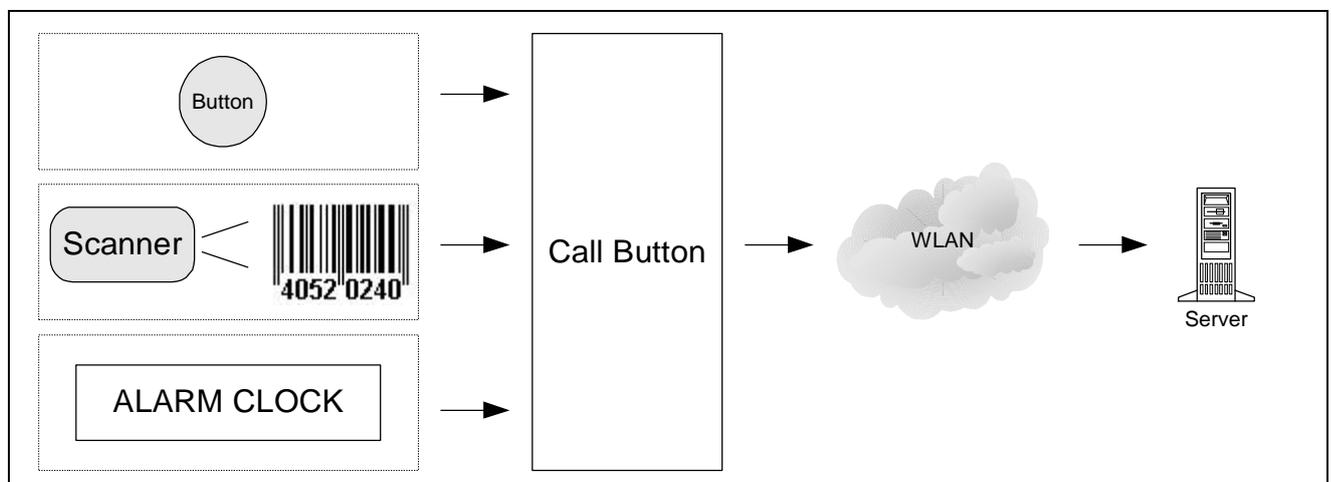


Abb. 2.2 – Melde-Ereignisse an den Server

Der Server verarbeitet das Ereignis und sendet eine Antwort zum Call Button. Dadurch wird z.B. ein Text auf dem Display angezeigt. Anschließend schaltet sich der Call Button wieder ab.

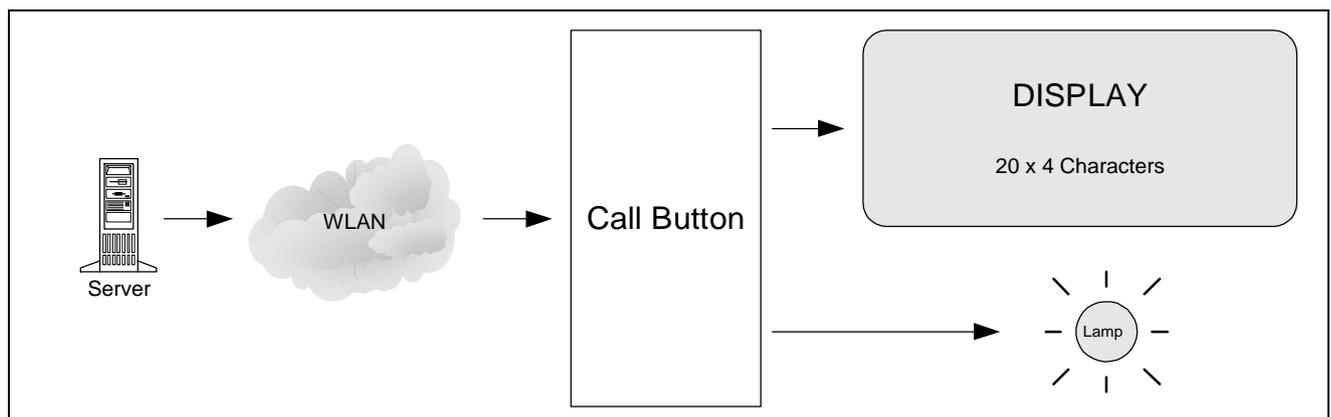


Abb. 2.3 – Antwort des Servers

2.2 Aufbau Hardware

Der tronic Call Button ist auf einfache Installation (ohne externe Versorgungsspannung) und lange wartungsfreie Betriebszeit ausgelegt. Er enthält eine langlebige Batterie und ein ausgefeiltes Powermanagement, so dass er – je nach Nutzung – mehrere Jahren ohne Batteriewechsel betrieben werden kann. Es wurde ein Betriebsprogramm (Firmware) geschrieben, das auf ein Standard-Betriebssystem verzichtet, um möglichst wenig Ressourcen auszukommen.

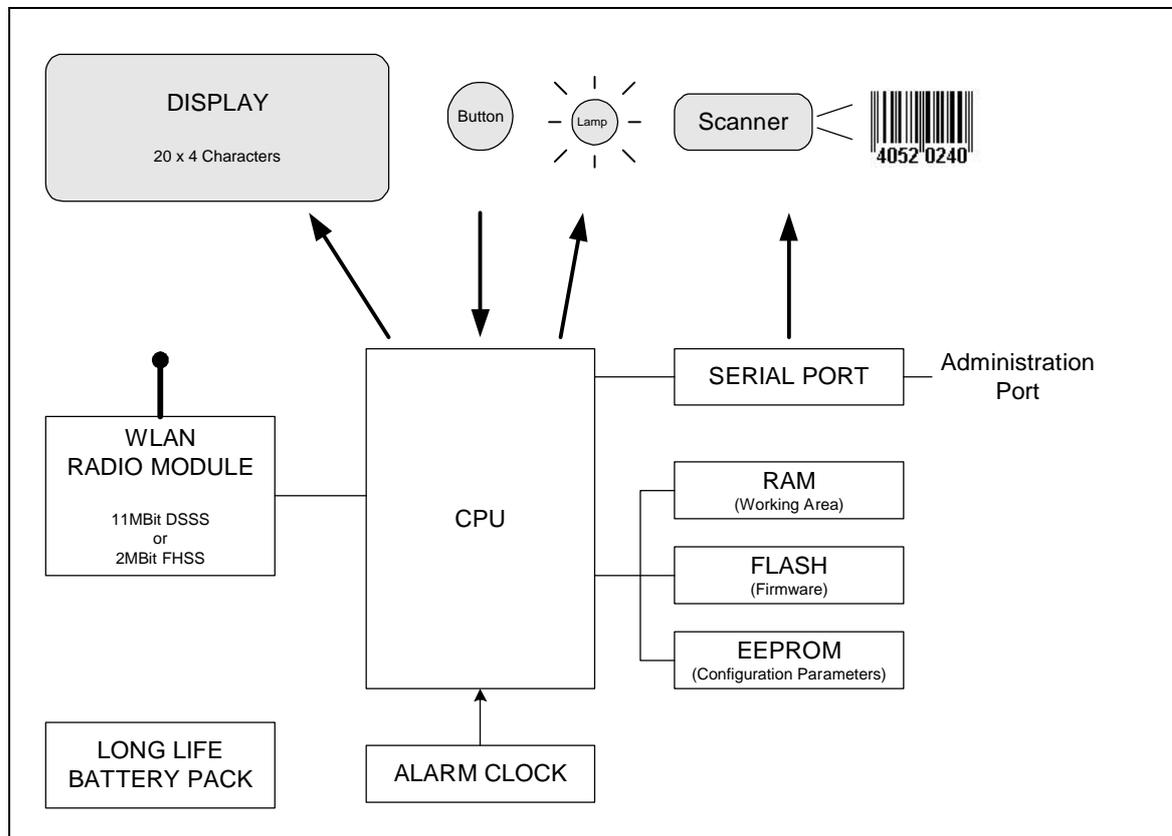


Abb. 2.4 – Blockschaltbild

2.3 Powermanagement

Der Call Button ist im Ruhezustand ausgeschaltet. Lediglich Display und Alarm Clock sind immer aktiv und belasten die Batterie mit nur wenigen Mikroampere. (Auf Anforderung des Servers kann außerdem die Blinksignalisierung im Ruhezustand eingeschaltet sein.)

Dadurch dass CPU und Funkmodul ebenfalls abgeschaltet sind, kann das Gerät nicht vom Server erreicht werden. Eine Netzwerk-Kommunikation kann also nur stattfinden, wenn das Gerät eingeschaltet wird. Es nimmt dann die Verbindung zum Server auf.

Folgende Ereignisse schalten den Call Button ein:

- Knopfdruck
- Barcode-Scanner
- Alarm Clock (Wecker)
- Administrations-Port (Laptop, etc.) ohne Netzwerkverbindung

Bei den drei Hauptereignissen wird sofort eine Verbindung zum Server aufgenommen und das Ereignis gemeldet. Nachdem die Antwort vom Server empfangen und verarbeitet wurde, schaltet sich der Call Button wieder ab.

2.4 Kommunikation

Bei der Entwicklung des Call Buttons war das oberste Designziel eine hohe Betriebszeit der eingebauten Batterie zu erreichen. Das geschieht insbesondere dadurch, dass das Gerät im Ruhezustand abgeschaltet ist. Der Server kann das Gerät also nicht seinerseits ansprechen. Die Initiative einer Kommunikation geht immer vom Call Button aus.

Die Kommunikation über das WLAN findet über das TCP/IP-Protokoll statt. Die Verbindung zum Server wird immer durch den Call Button aufgebaut, und zwar wenn er durch eines der folgenden Ereignisse eingeschaltet wurde:

- Knopfdruck
- Barcode-Scanner
- Alarm Clock (Wecker)

Wenn der Call Button keine fest konfigurierte IP-Adresse besitzt, holt er sich zunächst eine dynamische IP-Adresse vom DHCP-Server im Netzwerk. Die dynamische Adresse wird vor dem Abschalten wieder freigegeben, so dass sie durch den DHCP-Server neu vergeben werden kann.

Es wird ein Datentelegramm mit der entsprechenden Meldung gesendet. In Antwort darauf kann der Server:

- einen neuen Display-Text senden
- angeben, wie lange das Gerät eingeschaltet bleiben soll (in Sekunden, bis zu 4 Minuten)
- die Blink-Signalisierung aktivieren (wahlweise „solange Gerät eingeschaltet“ oder „nachdem Gerät abgeschaltet“)
- die nächste Weckzeit (in Minuten, bis zu 4 Stunden) festlegen
- die Konfiguration des Geräts ändern
- ein Firmware-Update durchführen

Vor dem Abschalten wird die TCP-Verbindung durch den Call Button wieder abgebaut.

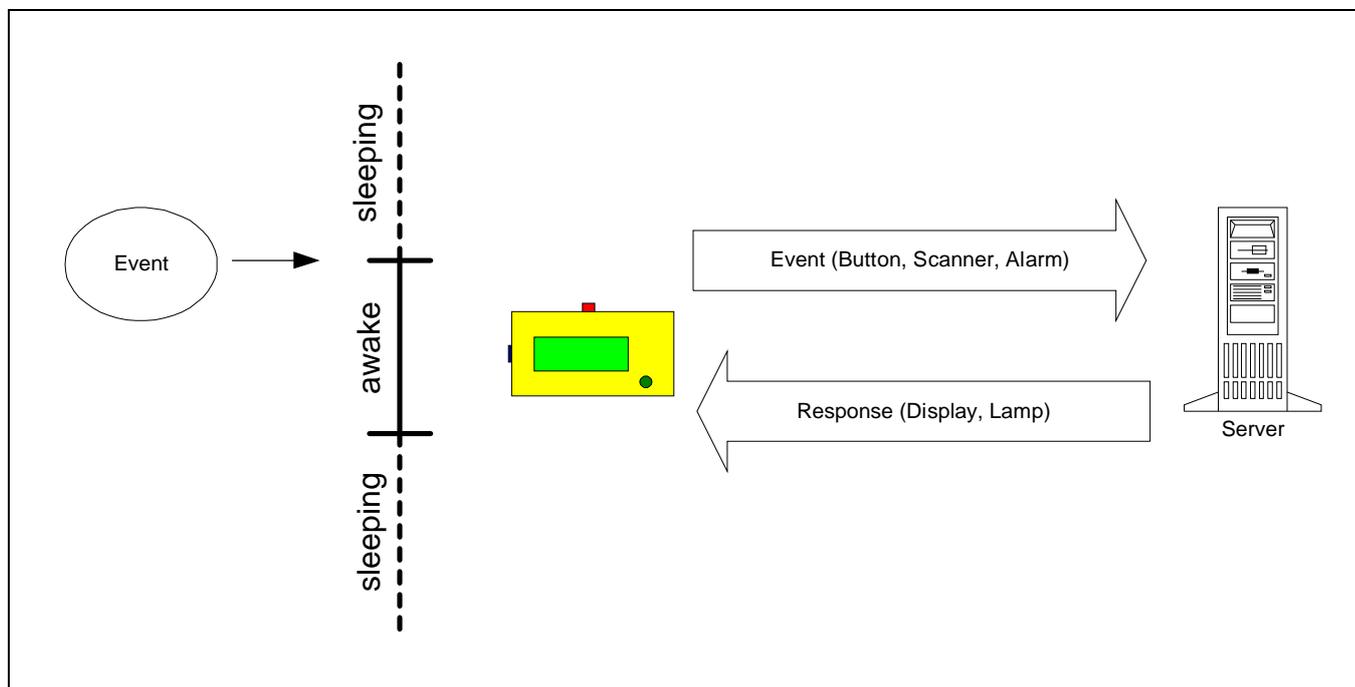


Abb. 2.5 – Normaler Kommunikationsvorgang

Die minimale Einschaltdauer hängt also von der Dauer der Netzwerkkommunikation und der Reaktionszeit des Servers ab.

Der Server kann in seiner Antwort bestimmen, dass das Gerät nicht sofort abschaltet (Anbleibzeit 0 bis 255 Sekunden). Die TCP-Verbindung wird aufrechterhalten und der Server kann weitere Telegramme zum Call Button senden, z.B. um das Display noch einmal zu ändern. Beispielsweise kann der Server in seiner ersten Antwort z.B. „Bearbeite Vorgang“ ausgeben, die interne Verarbeitung des Call-Ereignisses durchführen und anschließend, während der Call Button noch aktiv ist, das Ergebnis der Verarbeitung auf dem Display ausgeben lassen.

Ebenso ist es möglich, während der Anbleibzeit ein Firmware-Update durchzuführen oder die Konfiguration zu ändern.

Es ist jedoch zu bedenken, dass während der Anbleibzeit der Call Button der Batterie wesentlich mehr Strom entnimmt. Im Sinne einer hohen Betriebszeit der Batterie ist die Anbleibzeit also auf ein Minimum zu reduzieren.

Der Server kann ebenso die Weckzeit (1 bis 255 Minuten) bestimmen. Die im Call Button eingebaute Alarm Clock erzeugt nach dieser Zeit ein neues Ereignis, das dem Server gemeldet wird. Das Gerät kann somit veranlasst werden, regelmäßig beim Server anzufragen, ob es einen neuen Zustand im Ablauf gibt, also z.B. die Anzeige geändert werden soll.

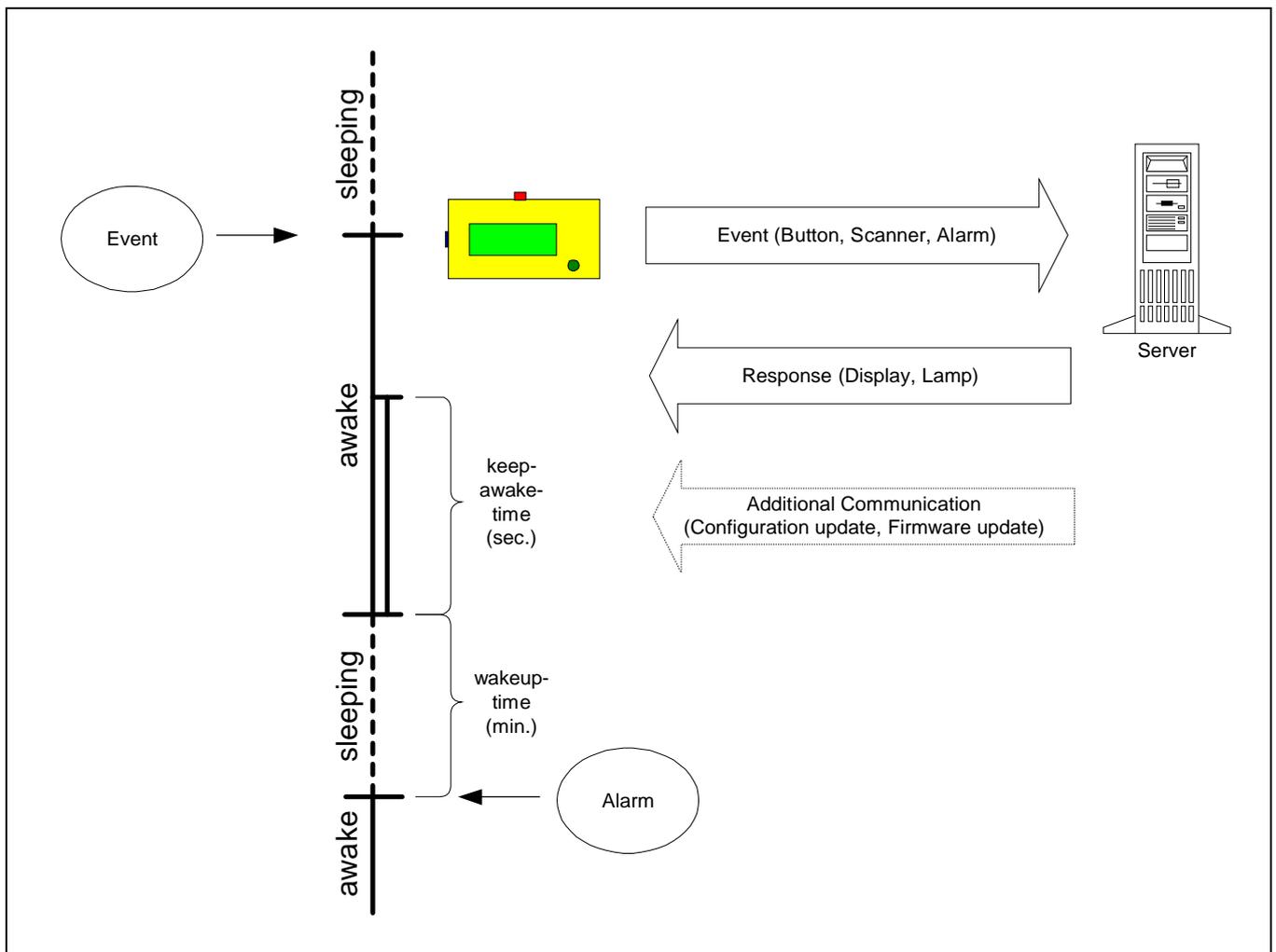


Abb. 2.6 – Kommunikationsvorgang mit vom Server übergebenen Zeiten

3 Kommunikationsserver

3.1 Kommunikationsserver

Der Kommunikationsserver ist für den Zugriff der Call Buttons auf die eigentliche Applikation und umgekehrt zuständig. Er stellt sich aus Sicht der Anwendung funktional (nicht technisch) transparent dar und ist projektunabhängig. Im Gegensatz dazu kann die Applikation auf die Erfordernisse eines konkreten Einsatzbereichs angepasst werden oder ohne Rücksicht auf die Besonderheiten des Call Buttons entwickelt werden. Die Administration der Call Buttons wird durch den Kommunikationsserver vorgenommen und von der Anwendung ferngehalten.

Der Kommunikationsserver führt die direkte Kommunikation mit den Call Buttons durch und speichert die aktuellen Zustände in einem Puffer. Er besitzt außerdem Schnittstellen zu unterschiedlichen Clients. Die wichtigste Schnittstelle ist natürlich die zur Applikation. Außerdem können diverse Administrations-Dienste angeschlossen werden.

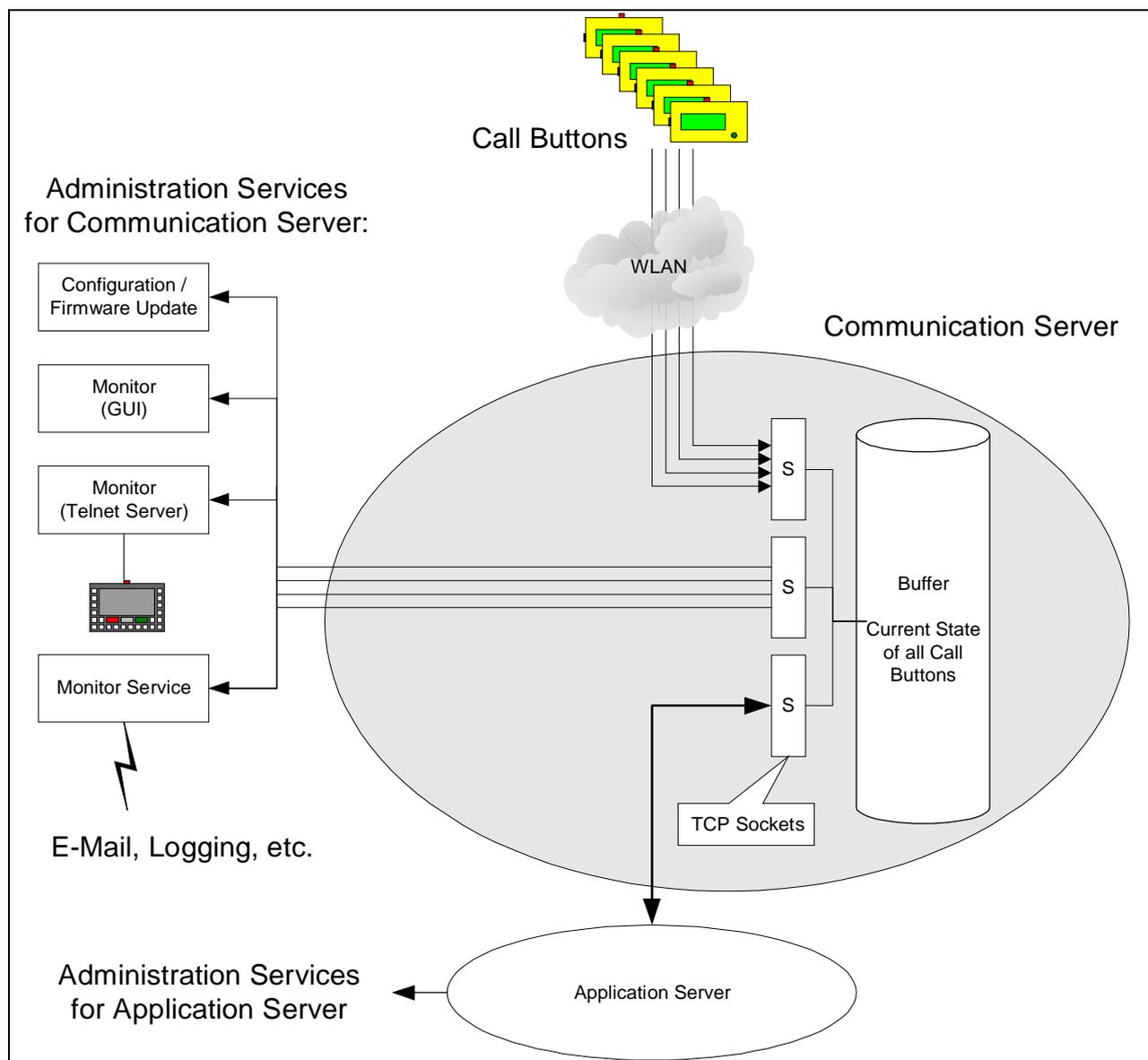


Abb. 3.1 – Kommunikationsserver

Der Kommunikationsserver läuft als Dienst (Service) unter Windows2000, WindowsXP und unter Windows NT 4.0-SP6 jeweils in der Professional oder Server Edition. Er enthält selbst keine Benutzeroberfläche, sondern wird über die Administrations-Clients gesteuert. Alle Schnittstellen (Call Buttons, Applikation, Administrations-Clients) werden über TCP/IP angesprochen, so dass der Kommunikationsserver und die Anwendung sowie die Clients verteilt auf unterschiedlichen Rechnern im Netzwerk laufen können.

3.2 Hilfsprogramme (Administration Services)

Die Administration Services sind Hilfsprogramme, die sich mit dem Kommunikationsserver verbinden und so das System konfigurieren und überwachen können.

3.2.1 Administration Client (GUI)

Mit dem Admin Client werden die Call Buttons und der Kommunikationsserver konfiguriert.

Der Admin Client initiiert Konfigurationsänderungen und Firmware-Updates für die Call Buttons. In einer Benutzeroberfläche können Einstellungen für die Geräte vorgenommen werden. Der Kommunikationsserver sorgt anschließend für die Synchronisation mit den Call Buttons und gewährleistet, dass alle Call Buttons auf den aktuellen Stand gebracht werden.

3.2.2 Monitor Client (Windows GUI)

Der Monitor-Client ist ein Windows-Programm und dient zur Überwachung der Call Buttons und der Kommunikation. Es verbindet sich mit dem Kommunikationsserver über TCP/IP und kann die aktuellen Zustände der Call Buttons in einer Benutzeroberfläche darstellen:

- an/aus
- aktueller Display-Inhalt
- Blinksignalisierung an/aus
- Batterieladung
- Zeitpunkt der letzten Kommunikation
- Zeitpunkt des nächsten Weckens
- Konfigurationsstand, Firmware-Version
- Netzwerk-Name, IP-Adresse, MAC, etc.
- Fehlerzustände

3.2.3 Mini-Monitor/-Admin Client (Telnet Server)

Ein einfacherer Monitor / Admin Client liegt als Telnet Server vor und läuft als Dienst (Service). Mit beliebigen Telnet-Clients (Unix, Windows, tronic Radio Terminal) kann der Kommunikationsserver angesprochen werden.

3.2.4 Monitor Service

Der Monitor Service läuft als Dienst (sinnvollerweise auf dem Rechner mit dem Kommunikationsserver) und überwacht die (Call Button- und Anwendungs-) Kommunikation.

Ein Mitschnitt (Historie) der Kommunikation kann zur späteren Analyse in eine Datei gesendet werden.

Bei bestimmten Ereignissen kann eine Meldung ausgelöst werden, z.B.:

- Batterie leer
- Fehlerzustände (z.B. Netzwerkfehler)
- Normale Call Button Ereignisse (z.B. Knopfdruck)
- Nicht-Melden eines Buttons innerhalb eines festgelegten Zeitraums

Als Meldung kann ein Protokoll geschrieben oder eine E-Mail gesendet werden. Der Monitor Service kann mittels eines weiteren Windows-Programms mit Benutzeroberfläche konfiguriert werden.

3.2.5 Call Button Konfiguration mit FBConfig

Für Konfiguration und Firmware-Update von Call Buttons steht ein vom Kommunikationsserver unabhängiges Programm zur Verfügung. Damit können einzelne Call Buttons per Laptop über die Serielle Schnittstelle eingerichtet werden.

Die Konfiguration kann in einer Datei abgespeichert werden und zu weiteren Call Buttons übertragen werden. Die Datei lässt sich mithilfe des Admin Clients auch per Kommunikationsserver auf alle Call Buttons verteilen, sofern die Netzwerkverbindung besteht.

4 Line-Feeding-Applikation

Die erste Anwendung für das Call System ist eine Line-Feeding-Applikation. Die Funkbuttons sind an unterschiedlichen Standorten der Produktionslinie einer Fabrik installiert und sollen auf Knopfdruck eine Materialanforderung melden. Aufgrund einer in der Anwendung konfigurierten Zuordnung wird ermittelt, welches Material gewünscht ist und ein Auftrag in eine Auftragswarteschlange gestellt. Aufträge in dieser Warteschlange werden an ein Telnet-Terminal (z.B. Funkterminal beim Flurförderfahrzeug) erteilt und die Durchführung überwacht. Auf der Buttonanzeige wird der jeweilige Bearbeitungsstatus dargestellt.

4.1 Applikationsserver

Der Applikationsserver übernimmt die Zuweisung der Aufträge zu den verfügbaren Förderfahrzeugen. Er ist wie der Kommunikationsserver als Dienst (Service) installiert und besitzt selbst keine Benutzeroberfläche. Er kann aber muss nicht auf demselben Rechner laufen, auf dem der Kommunikationsserver installiert ist.

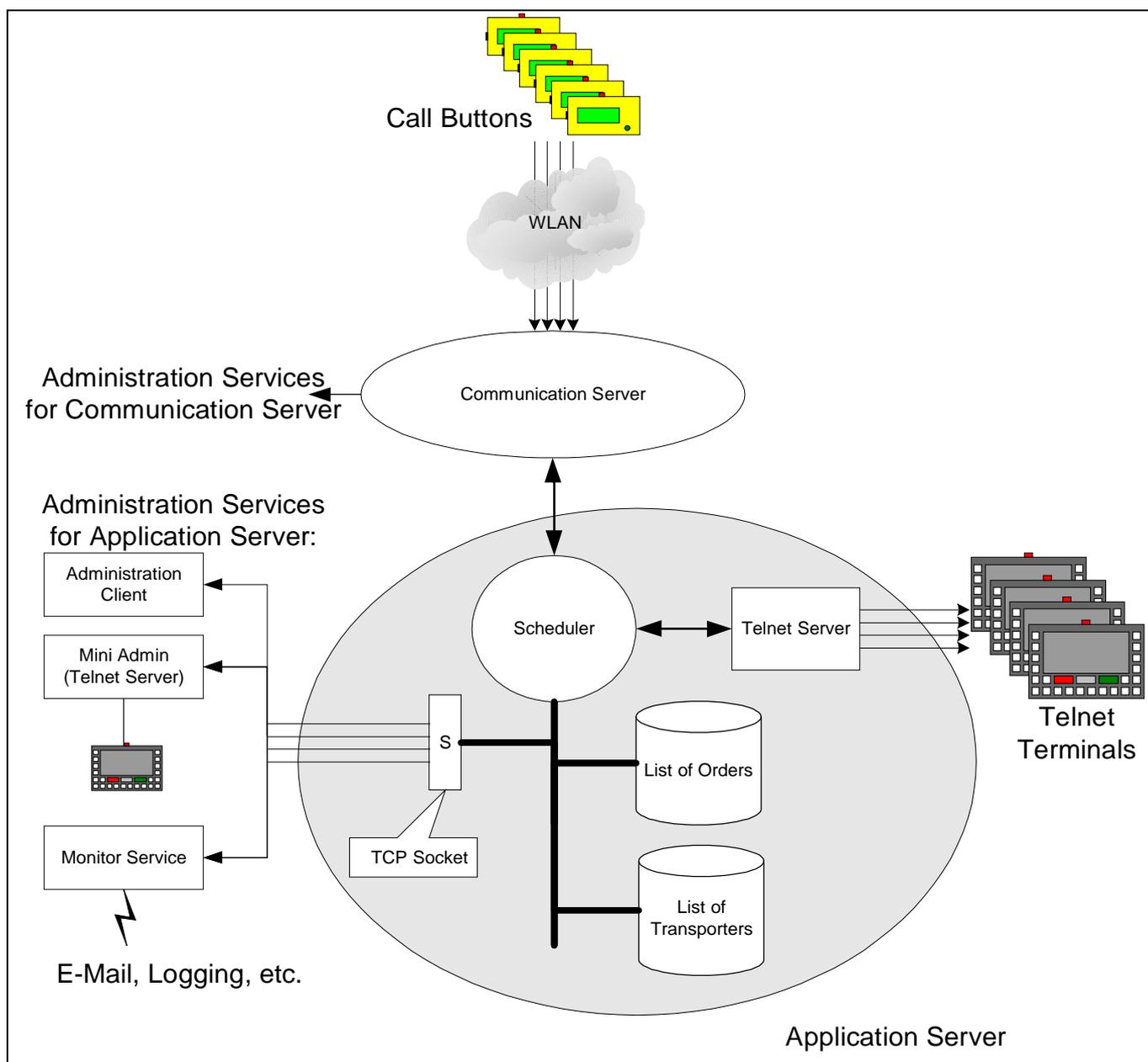


Abb. 4.1 – Applikations-Server

Die Terminals der Förderfahrzeuge verbinden sich mit dem in der Applikation eingebauten Telnet-Server und melden sich am System an. Der Scheduler vermittelt die Anforderungen der Call Buttons an die Förderfahrzeuge.

Die Administration der Applikation geschieht über ein externes Programm mit Benutzeroberfläche. Es verbindet sich über TCP/IP mit dem Applikationsserver. Über das Programm können die aktuellen Zustände der Transporteinheiten (Förderfahrzeuge) und der Aufträge (Jobs) angesehen und bearbeitet werden, usw.

Der Ablauf einer Materialanforderung ist in der folgenden Abbildung dargestellt. Dabei löst ein Call Button einen Auftrag aus und fragt per Weck-Ereignis gelegentlich nach dem aktuellen Stand der Bearbeitung.

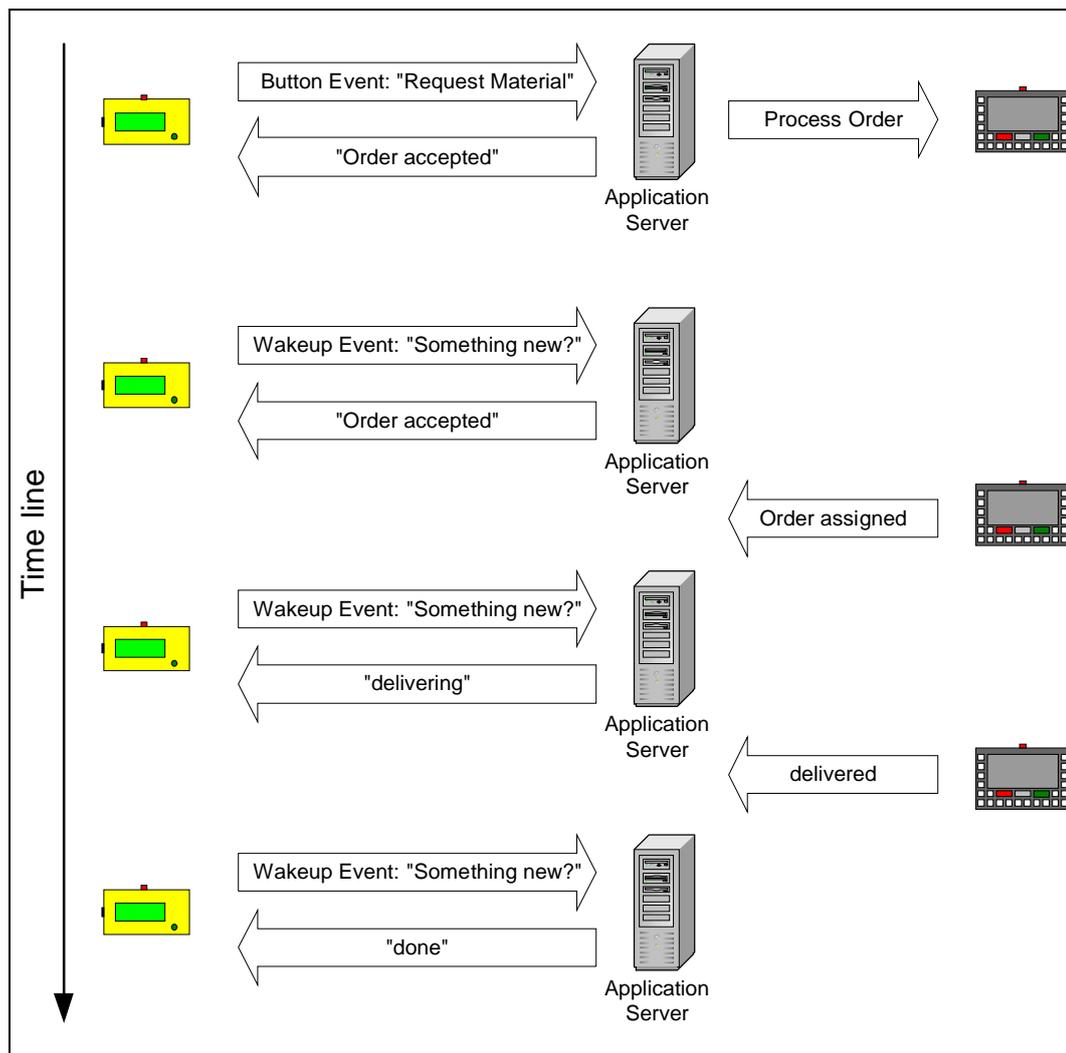


Abb. 4.2 – Ablauf einer Materialanforderung

Die folgenden Abbildungen zeigen die einzelnen Stufen einer Auftragszuweisung im Applikationsserver. Dargestellt wird die Auslösung durch den Call Button, die Zuweisung an eine Transporteinheit und die jeweiligen Rückmeldungen an den Call Button.

Dabei wird der dargestellte Call Button über den Kommunikationsserver angesprochen. Der Kommunikationsserver ist im Gegensatz zu den Call Buttons ständig ansprechbar und sorgt in einem asynchronen Prozess dafür, dass die jeweils aktuelle Rückmeldung auf dem Display angezeigt wird, wenn der Call Button aktiv ist oder sobald er sich meldet.

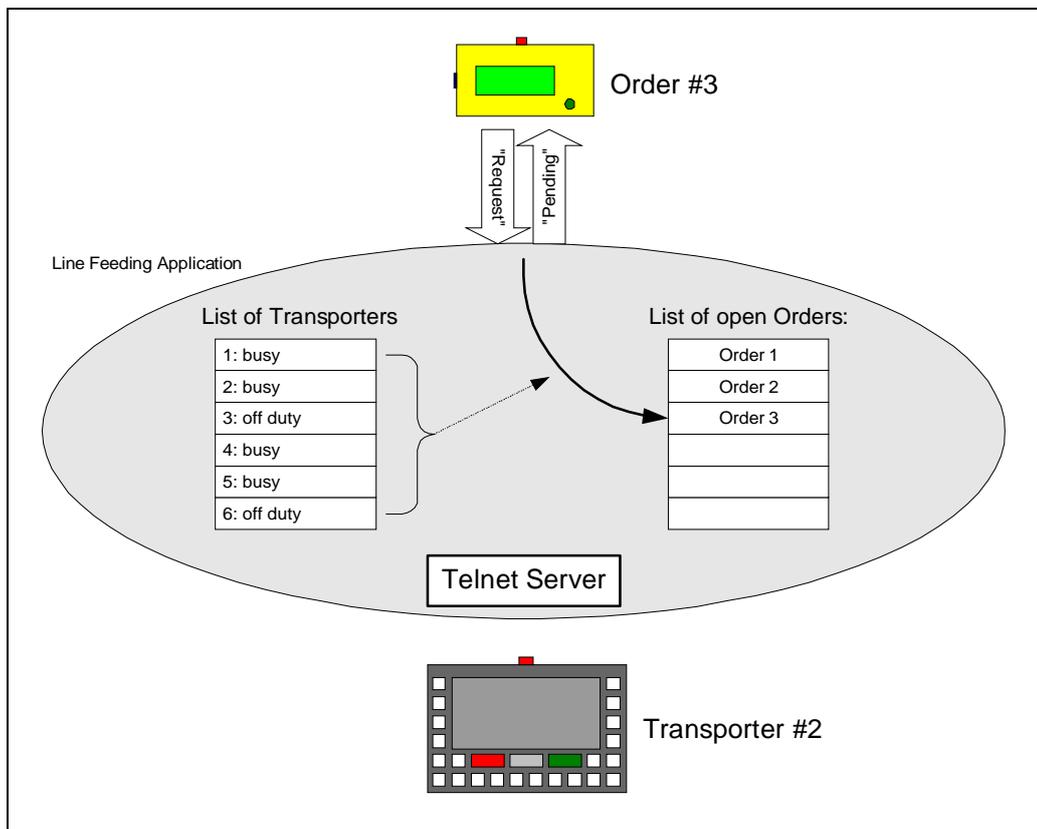


Abb. 4.3 – Auftragszuweisung 1. Stufe: Auftrag wird ausgelöst

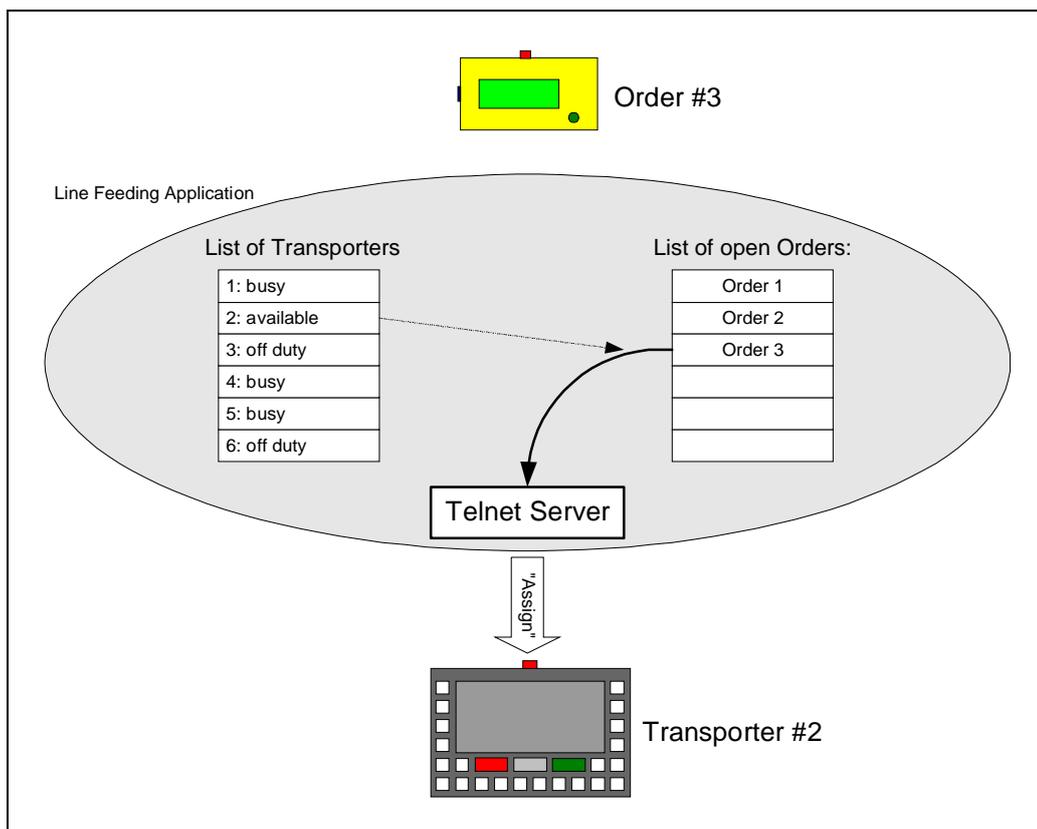


Abb. 4.4 – Auftragszuweisung 2. Stufe: Auftrag wird zugewiesen

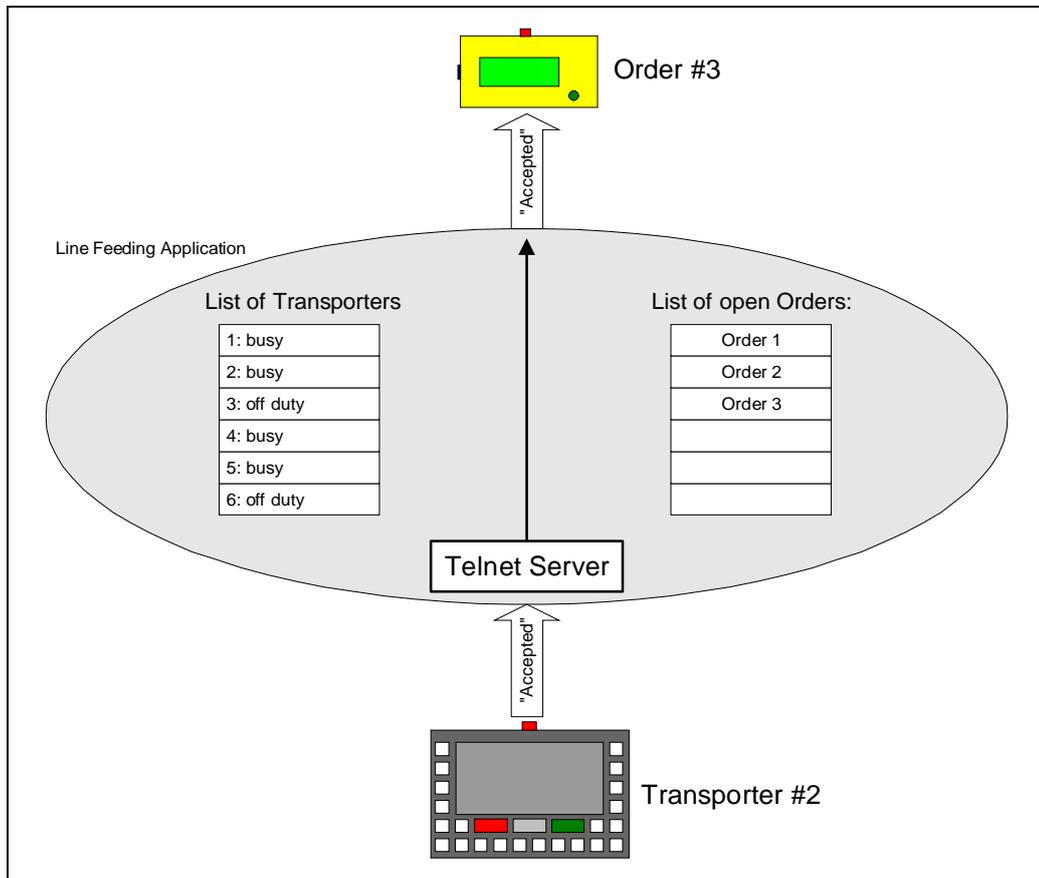


Abb. 4.5 – Auftragszuweisung 3. Stufe: Auftrag wird akzeptiert

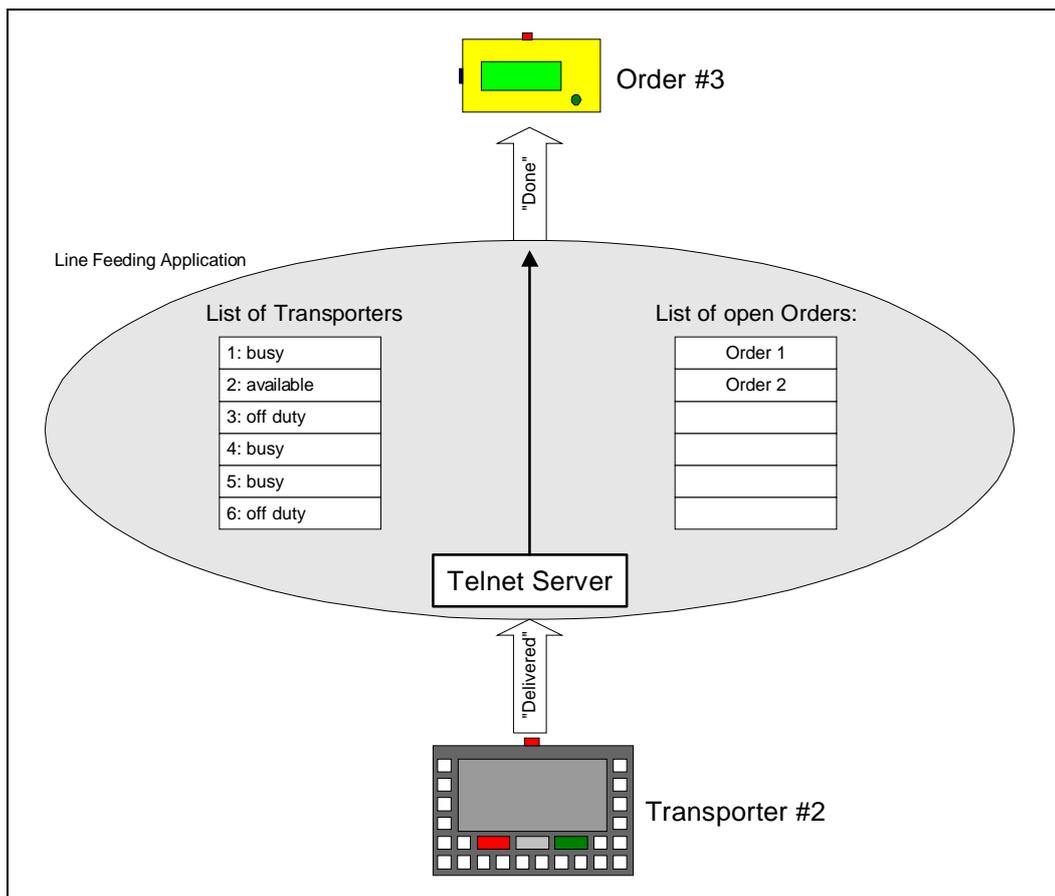


Abb. 4.6 – Auftragszuweisung 4. Stufe: Auftrag wurde erledigt

4.2 Konfiguration

Zum Betrieb der Line-Feeding-Applikation müssen die Stammdaten eingerichtet werden:

- Zuordnung der Call Buttons (User) zu den Standorten und den Materialbedarf (Teilenummer, Menge)
- Pro User: Priorität der Auftragszuordnung, Zeitlimit für Durchführung
- Zuordnung der User zu Benutzergruppen (Mandanten)
- Einrichtung der Transportfahrzeuge (Standort; Attributen wie Kapazität, Spezialeinrichtungen, etc.)
- Zuordnung der Fahrzeuge zu Benutzergruppen
- etc.

4.3 Hilfsprogramme (Administration Clients)

Die Administrations-Clients sind Hilfsprogramme ähnlich denen des Kommunikationsservers. Sie verbinden sich mit dem Applikationsserver und erlauben die Konfiguration und Überwachung des Systems.

4.3.1 Admin Client (Windows GUI)

Der Admin Client stellt eine Windows-Benutzeroberfläche zur Verfügung, mit der der Applikationsserver eingerichtet werden kann. Dazu gehört z.B.:

- Zuordnung von Call Buttons zu Standorten
- Material (Teilenummer), Menge und Lagerort
- Priorität der Auftragszuweisung
- Maximale Zeitspanne, bevor Fehler ausgelöst wird
- Standardtexte / -masken für die Anzeige des Auftragsstatus auf dem Call Button Display

Die Call Buttons können aus Sicht des Standorts oder der Teilenummer überwacht werden (aktueller Zustand wie beim Monitor-Client des Kommunikationsservers)

Außerdem können mit diesem Programm die Jobs (Materialanforderungen) und deren Zuweisung an die Förderfahrzeuge überwacht und manipuliert werden. Damit können Aufträge gelöscht oder wieder freigegeben werden.

4.3.2 Mini-Admin Client (Telnet Server)

Ein einfacher Admin Client liegt als Telnet Server vor und läuft als Dienst (Service). Mit beliebigen Telnet-Clients (Unix, Windows, tronic Radio Terminal) kann der Applikationsserver angesprochen werden.

4.3.3 Monitor Service

Der Monitor Service läuft als Dienst (sinnvollerweise auf dem Rechner mit dem Applikationsserver) und überwacht die Auftragsabwicklung.

Ein Mitschnitt (Historie) der Aufträge und Zuweisungen kann zur späteren Analyse in eine Datei gesendet werden.

Bei bestimmten Ereignissen kann eine Meldung ausgelöst werden, z.B.:

- Aufträge können nicht zugewiesen werden (Zeitüberlauf)
- Fehlerzustände

Als Meldung kann ein Protokoll geschrieben oder eine E-Mail gesendet werden. Der Monitor Service kann mittels eines weiteren Windows-Programms mit Benutzeroberfläche konfiguriert werden.

Der Monitor Service kann weitere Informationen zur statistischen Auswertung sammeln (z.B. Bestellungen je User oder Benutzergruppe je Zeiteinheit).

5 Technik

5.1 Plattform

5.1.1 Server

Der Kommunikationsserver und der Line-Feeding-Applikation laufen als Dienst (Service) unter Windows2000 / XP und NT4.0/SP6. Sie können auf getrennten Rechnern im Netzwerk installiert werden.

5.1.2 Windows-Clients

Die unterstützenden Windows-Programme mit Benutzeroberfläche zur Administration der Server laufen unter Windows98/2000/XP/NT4. Sie können auf Arbeitsplatzrechnern mit Netzwerkzugriff auf die entsprechenden Server installiert werden. Sie sind für mehrere Sprachen verfügbar.

5.1.3 Telnet-Server

Die Telnet-Server zur Administration laufen unter Windows2000 / XP und NT4.0/SP6 mit Netzwerkzugriff auf die entsprechenden Server. Sinnvoll ist, sie auf denselben Rechner wie der entsprechende Server zu installieren. Sie können mit Telnet-Terminals (VT100) angesprochen werden.

5.2 Zugriffsschutz

Alle Client-Programme sind mit einem Passwort versehen, so dass unabhängig von den Netzwerkrechten keine unautorisierten Konfigurationsänderungen vorgenommen werden können.

5.3 Lizenzierung

Die Anzahl der vom Kommunikationsserver anzusprechenden Call Buttons ist entsprechend zu lizenzieren.

Beim Applikationsserver ist die Anzahl der Call Buttons und der Terminals zu lizenzieren.

Eine Erweiterung der Lizenzen ist auch nachträglich möglich.

5.4 Weitere Applikationen

An dem System aus Call Buttons und Kommunikationsserver können auch andere, projektspezifische Anwendungen angeschlossen werden. Die Applikations-Schnittstelle des Kommunikationsservers ist dokumentiert.