PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : **H04Q 7/24, H04L 12/24, G06F 17/30**	**A1**	**(11) International Publication Number:** **WO 00/54525** **(43) International Publication Date:** 14 September 2000 (14.09.00)

(21) International Application Number: PCT/AU00/00174

(22) International Filing Date: 10 March 2000 (10.03.00)

(30) Priority Data:
PP 9117 10 March 1999 (10.03.99) AU

(71) Applicant *(for all designated States except US)*: RETRIEVER COMMUNICATIONS [AU/AU]; Ground floor, 45 Booralie Road, Terrey Hills, NSW 2084 (AU).

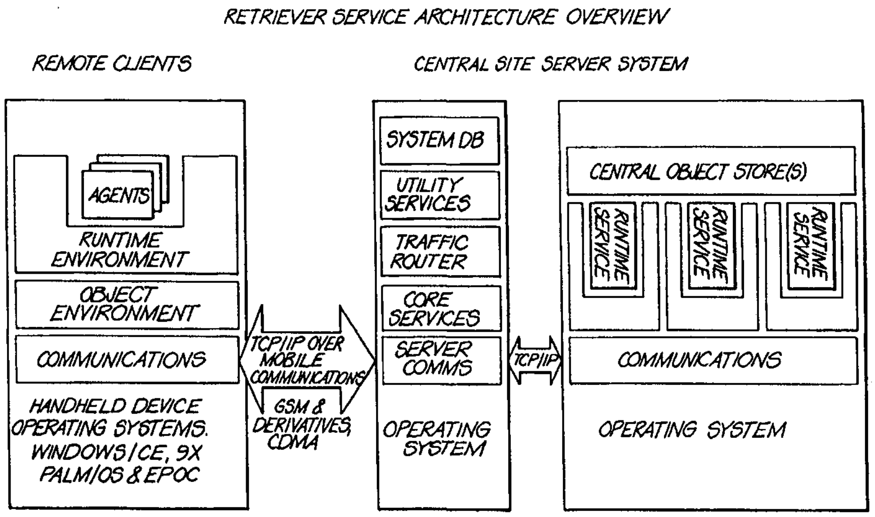
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: BRITTAIN-WHITE, Mary [AU/AU]; 983 Pittwater Road, Collaroy, NSW 2097 (AU). KOCH, Peter [AU/AU]; 19/98 Dee Why Parade, Dee Why, NSW 2099 (AU).

(74) Agent: F B RICE & CO.; 605 Darling Street, Balmain, NSW 2041 (AU).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, 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, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: MOBILE APPLICATION SERVICE



(57) Abstract

A first aspect is a centralised server, or software to program a centralised server, for a mobile application service system, where the server is programmed to create and maintain separate databases for respective mobile clients to hold client knowledge. A second aspect is a client runtime environment, or software to program a runtime environment. There are also methods for operation. They involve free form database objects, known as frame objects, which hold client data in data fields and are replicated in client computers. Component based application modules, known as agent objects, which the server despatches to respective clients, as required, to create or manipulate information in the frame objects. The server and runtime environment are programmed such that, after a frame object has been changed a new edition of the frame object, known as a delta frame object, is created which contains only the data fields that have been changed. Synchronisation of the server and the mobile client involves the exchange of edition information between them so that a determination can be made of which delta frame objects need to be aggregated and transmitted in order to bring them into conformity. Then transmission of the determined delta frame objects achieves conformity.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

Title

MOBILE APPLICATION SERVICE

Technical Field

This invention concerns a mobile application service. In particular the invention concerns the server and remote client, and their methods of operation, in a system for providing a mobile application as a service, to users.

Summary of the Invention

In a first aspect the invention is a centralised server, or software to program a centralised server, for a mobile application service system, where the server is programmed to create and maintain separate databases for respective mobile clients to hold client knowledge in the form of:

free form database objects, known as frame objects, which hold client data in data fields and are replicated in client computers; and

component based application modules, known as agent objects, which the server despatches to respective clients, as required, to create or manipulate information in the frame objects;

and where the server is programmed such that, after a frame object has been changed a new edition of the frame object, known as a delta frame object, is created which contains only the data fields that have been changed; and synchronisation of the server and the mobile client involves the exchange of edition information between them so that a determination can be made of which delta frame objects need to be aggregated and transmitted in order to bring them into conformity, then transmission of the determined delta frame objects achieves conformity.

In a second aspect the invention is a method of providing a mobile application service system, comprising the steps of:

creating and maintaining at a server, separate databases for respective remote clients to hold client knowledge in the form of:

free form database objects, known as frame objects, which hold client data in data fields and are replicated in client computers; and component based application modules, known as agent objects, which the server despatches to respective clients, as required, to create or manipulate information in the frame objects;

and where, after a frame object has been changed, creating a new edition of the frame object, known as a delta frame object, which contains only the data fields that have been changed; and synchronising the server and the client by exchanging edition information between them, determining from the exchange which delta frame objects need to be aggregated, aggregating those delta frame objects, and transmitting the aggregated delta frame objects in order to bring them into conformity.

In a third aspect the invention is a client runtime environment, or software to program a runtime environment, where the runtime environment is programmed to hold: free form database objects, known as frame objects, which hold client data in data fields and are replicated in a remote server, and component based application modules, known as agent objects, received from the server despatches, as required, to create or manipulate information in the frame objects;

and where the runtime environment is programmed such that, after a frame object has been changed a new edition of the frame object, known as a delta frame object, is created which contains only the data fields that have been changed; and synchronisation of the client with the server or another client involves the exchange of edition information between them so that a determination can be made of which delta frame objects need to be aggregated and transmitted in order to bring them into conformity, then transmission of the determined delta frame objects achieves conformity.

In a fourth aspect the invention is a method of providing a runtime environment, comprising the steps of:

programming the runtime environment to hold: free form database objects, known as frame objects, which hold client data in data fields and are replicated in a remote server, and component based application modules, known as agent objects, received from the server despatches, as required, to create or manipulate information in the frame objects;

and where, after a frame object has been changed, creating a new edition of the frame object, known as a delta frame object, which contains only the data fields that have been changed; and synchronising the client with the server or another client by exchanging edition information between them, determining from the exchange which delta frame objects need to be aggregated, aggregating those delta frame objects, and transmitting the aggregated delta frame objects in order to bring them into conformity.

By this arrangement and technique it may be possible to minimise the volume of data transfer required to maintain synchronisation between one or more remote client databases, where each client database holds several frame objects. Alternatively, it may be possible to minimise the volume of data transfer required to maintain synchronisation of a centralised, server, database holding an amalgamation of all frame objects which have been replicated to the server database from client databases.

The server and client databases cooperate in the process of synchronisation to maintain editions of frame objects stored at the server and client databases. An edition represents a complete frame object when taken in isolation. An edition, when taken relative to a predecessor edition represents only the changes ("delta elements") to that predecessor edition. The edition history may be arbitrarily long (i.e. the number of editions kept) although for data storage conservation reasons edition history may be trimmed without loss of integrity of the whole frame object. The tradeoff in this case will be that the whole frame object would need to be sent in any request for an edition which has been trimmed since the change information required to enable extraction of the delta elements will have been removed. Generally, Objects are held at the server and client in persistent storage.

All the objects are advantageously platform independent. Frame objects can have data fields ("slots") arbitrarily added, changed or deleted.

Conventional compression schemes may also be employed to further improve the efficiency of data transfer.

The computer supporting the server database and the computers supporting the client databases will be connected via communications networks which are either charged on a time connected basis, or, a quantity of data transferred basis. In either case it is to the commercial benefit of the user to minimise the data traffic such that telecommunications costs are reduced.

Programming and administration tools may be provided for the system.

The system may be designed to offer reliable uninterrupted operations. The design may be flexible and broad enough to meet both current and future subscriber requirements. The system may also minimise telecommunications traffic costs and cater for the typical mobile user who is "occasionally" connected to the service.

The central server infrastructure may support multiple companies operating simultaneously and with concurrent end-user sessions. It may be able to be rapidly and easily customised to meet an individual subscriber company's needs. It may be capable of adding or changing customised applications on the service without compromising the operating stability and reliability for existing subscribers. It may also provide users with an environment closely representing their "legacy" paperwork processes. The system may interface with existing host computing systems to extend the functionality of a company's IT as seamlessly as possible. It may also maximise operational cost efficiency by sharing central server infrastructure over the entire subscriber base. The service may support clients operating in "occasionally connected" mode to provide continuous functionality for mobile users or fixed clients with restricted communications continuity. It may minimise data transfer between clients and the server by only transferring those elements of data structures that have changed at either end. It may support mechanisms to stringently validate the integrity of data transferred between the server and clients. It may be capable of being localised to enable deployment in different international territories

The architecture enables the implementation of a reliable multi-subscriber mobile application service. An innovative use of object technology and an application abstraction layer enables the system to run generalised software at the server (hub) and yet be fully customable to the unique needs of different subscriber companies.

Brief Description of the Drawings

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

Figure 1 is a schematic diagram of the tiers of the architecture of the service example.

Figure 2 is a schematic diagram of the architecture of the central service system.

Figure 3 is a schematic diagram of the client architecture.

Figure 4 is a schematic diagram of the communications structure assumed for the service.

Best Modes of the Invention

The system architecture 1 of the service system is implemented as a multi-tiered client-server computing model, as illustrated in Figure 1. There is a central site server system 2, and remote clients 3. There are a number of tiers of the architecture.

A client-server communications layer 4 supports network traffic authenticated access, utility services and programmable (custom) services. The programming model (client-server-service) relies on the concept of software services running in conjunction with the server with which clients interact.

A system-wide object programming model 5 supporting platform independent objects, object manipulation primitives (methods) and mechanisms for remote method invocation between clients and the server. An application "runtime" environment 6 provides support for advanced object operations such as persistence (caching), queries, transfer, synchronisation, etc.

Component-based application modules 7 are dispatched to execute in the client runtime environment 6. The applications 7 are called agents. Agents 7 are objects able to create or manipulate information, which will subsequently be exchanged with the server system 2, deposited in an object store 8, and routed to their next destination (client). To an end user, agents might appear as forms and list items that display on the client device screen.

The system also has a number of other important functional blocks. These are: a System Development Kit (SDK) that provides a set of programming tools that enable the rapid development of agents and other data definitions used to define an application. And a set of administrative tools used in the operation and maintenance of the overall system.

The lower layer consists of an object-oriented client-server system that provides data communications, authenticates connections, routes data traffic, logs events, and provides a variety of other utilities (e.g., Internet mail and paging system interfaces).

The client-server system implements the object model and object primitives used extensively by the upper layers of the architecture. The object model is built around clients interacting with services, which are exposed to them through the server. Clients create remote objects of defined types in services and send method calls to those objects for the server to execute on their behalf.

In single-purpose application systems, the services consume a large proportion of the application specific code. However, this service system provides services that support object manipulation for the runtime environment executing in the clients. Subscriber-specific application code executes exclusively in the client runtime environment 6.

Applications are abstracted from the basic client-server system by a runtime layer. The client side runtime layer 6 implements a set of interfaces and functionality required for the execution of application agents 7 at a client. Agents 7 are stored centrally 8 on the server and dispatched to a client as required. As far as an agent 7 is concerned, all the resources and data that it requires to operate exist in the client runtime environment 6. Any requests that the client runtime cannot fulfil directly for an agent will be dealt with by having the runtime 6 "contact" the server 2. This can happen invisibly or the agent 7 can be notified of a connection requirement.

The central server system 2, or hub, is the transportation and distribution centre for the entire system. The hub 2 enables agents 7 to travel to clients 3 in order to carry out their intended tasks. Upon connection, clients 3 declare their platform type to the hub 2 which, in turn, dispatches agents 7 that are compatible with that client's computer platform. The hub 2 is solely responsible for transporting and storing objects; it remains independent of any particular subscriber application.

The hub 2 is designed to support multiple subscriber companies using the system simultaneously and multiple concurrent client sessions. The services at the hub are available to all users.

Figure 2 shows an overview of the hub architecture.

The hub 2 moves all application data to and from clients. The hub is configured for a new subscriber company by allocating a database 10 to a subscriber company and then depositing agents 7 and their accompanying data into that database10. The software "knowledge" for a particular subscriber company is held entirely by the company specific agents and their accompanying data and view definitions.

Administration and monitoring of the overall system is carried out by client computer(s) with "administrator" privileges. Administrator clients are independent of the application system and interact directly with the client-server and runtime systems.

Administrative clients are used to manage functions such as adding or removing subscriber end users, depositing new versions of agents onto the system for use in a particular subscriber application, viewing system logs to trace abnormal events or faults, etc.

Agents 7 characterise the behaviour of a client 3. Agents 7, in conjunction with the data store 10, are the primary means of customising the system. Agents7 stored at the hub 2 may contain code for execution on a number of different client platforms. The hub 2 will choose the appropriate code version to dispatch in an agent to a client 3. It is able to do this because clients 3 declare their platform type to the hub when they connect in.

Neither the server nor client runtime layers 6 have any specific knowledge of the tasks that an agent implements for a particular subscriber company. The runtime environment deals with objects only. Agents are uniquely programmed to implement the "business rules".

Agents are dispatched from the hub to clients automatically and are held in persistent storage in a client until replaced or purged. Therefore, the hub may respond to a client's request for an agent and attached data by sending the data only, since the agent may already reside in memory. Agents may co-exist in clients with more versions of themselves. This enables multiple versions of "agent data" to reside in the system and still be processed by the agent of the correct (or older) version if required. This feature is important where large amounts of historical data may be held on the system. Older versions of data may need to be handled by a matching version of an agent. In this case agents and agent data of older versions must be able to co-exist.

Agents "carry around" the data that they require by referencing data objects. The actual task of transporting the data is a function of the object store and the routing and transfer services. Consequently, agents never need to concern themselves with how they or the data they require travels.

The major elements of the client 3 architecture are shown in Figure 3.

The scope of an agent's 7 interaction with the client operating system or connected devices is unrestricted. It is, therefore, possible to implement any required interface at a client. Client computers may interact with human operators using an agent that has a Graphical User Interface (GUI). Client computers can also act as a gateway to other computers or IT systems simply by creating the appropriate agent interface. Standard agents can be constructed for use via published programmatic interfaces as ODBC and those that are delivered with the Operating systems and database software. Other standard agent interfaces for integrating with customer information systems such as SAP can be readily constructed.

The system is designed to enable clients to operate in off-line (not connected) mode. The runtime environment attempts to satisfy all agent requests without connecting. The object store caching mechanism captures data current to the last connection between the hub and a client. The off-line mode of operation is used heavily for mobile clients; however, the use of the "occasionally connected" mode of operation is not restricted to use by mobile devices.

With the recent emergence of intelligent mobile phone handsets, this class of device is a potential candidate for a mobile client computer. The screen area, memory and processing power of these devices is generally limited to the extent that they are not candidates for a standard client. However, by using a standard client as an intermediate interface for these types of devices, it would be possible to support such simple mobile clients on the system. A single interface client could act as a proxy for multiple simple clients. Simple clients would generally be restricted to receiving schedule (job) data, acknowledging receipt of schedule data, submitting time use information. Simple clients could also perform well in the receiving of management information from the system so that business performance information could be conveyed to field based management.

An agent implements its GUI with the aid of View Definitions. A View Definition is yet another object and defines how a particular GUI operates on a client device. This is important where the same agent may execute on a variety of devices with differing screen sizes and input systems (e.g. handheld vs desktop or keyboard vs pen). Therefore, view definitions provide convenient and consistent ways for agent data to be displayed and input on a system wide basis.

In order to have data formats universally understood across the entire system all data is described by data definitions. Data definitions are objects with a known internal structure which have the sole purpose of describing the format of other free form objects. Data definitions (DataDefs) have version information contained in them (just like agents). Consequently, multiple versions of the same basic data definition may exist in the system concurrently without causing concern (as you might expect this is actually true of all objects used in the application system).

Data definitions may reference other data definitions. Arbitrarily complex data structures can be constructed and moved around the system through this linking and embedding mechanism.

Each customised application will have its own set of data definitions which will be guaranteed to be unique within the system.

The object stores (hub and client) hold all subscriber specific data in the system. The hub object store is designed for fast operations on multi-object hierarchical structures. The object store only moves those objects that do not currently exist at the destination site (client or server). The object store further economises on data traffic by moving only those parts of objects that have changed. This process of partial data transfer is termed "synchronisation". Synchronisation plays an important role in minimising traffic to and from mobile clients where communications bandwidth is at a premium.

In more detail, when a frame object has been changed a new edition of the frame object, known as a delta frame object, is created which contains only the data fields that have been changed. Synchronisation of the server and the client involves the exchange of edition information between them so that a determination can be made of which delta frame objects need to be aggregated and transmitted in order to bring them into conformity. Then the determined delta frame objects are transmitted to achieve conformity.

An object based query system is built into the object store in order to enable content based searching and extraction of objects.

The hub object store caches all client data that passes through it. Clients also cache data but restrict this to items that are of concern to them. Therefore, the hub has a snapshot of information from clients current to their last connection. The caching of data at the hub enables the recovery of a particular client's state should it lose its local copy of data.

All data associated with each subscriber company is held in separate object stores at the hub.

Customisation to meet individual subscriber company requirements is carried out with the aid of a System Development Kit (SDK). The SDK generates agents, view definition and data definition objects. Agents, view definitions and data definitions created by the SDK are loaded into storage on the hub for distribution to client computers. The SDK supports the creation of derivative applications from base (template) applications. This substantially speeds the implementation of applications for specific companies. Derivative applications can co-exist in the system without conflict since all objects are assigned unique IDs.

The basic System requires mobile communications to support roaming connections from handheld clients, landline communications to support connections from office based clients and landline communications to support administration clients.

The communications structure assumed for the service is shown in Figure 4.

GSM is the prime wireless communications medium for handheld devices. This is well supported in Australia, Asia and Europe with the majority of GSM carriers having data capabilities in their networks. In order to increase the customer satisfaction with the GSM mobile connections there is "direct connect" of service servers to the GSM switch. Direct connect eliminates the need to exit the digital part of the GSM network in order to establish a data connection via analog modem to the server(s). This has the effect of reducing the connect time from a mobile handheld by approximately 20 seconds (analogue modem dialling/training time). The resulting connect time is then reduced to the TCP/IP communication setup and server login time.

Advances in the GSM system will soon see the introduction of GSM packet based services known as General Packet Radio Services (GPRS). Packet services enable a (virtual) continuous connection between the client and the hub with the subscriber being charged by data traffic rather than connection time.

The implementation of GPRS will improve the utility of the Service by enabling mobile clients to dramatically increase their communication "uptime" with the service. While ever a mobile user is in radio coverage GPRS will enable client to server communications to take place and a more continuous exchange of information between the mobile user and the hub. Alternative mobile data networks such as CDPD are supported by the Retriever system. CDPD is a packet based system supporting TCP/IP communications. It is enjoying increased coverage in the USA and offers similar features to GPRS but operates on an analogue system (operates over legacy AMPS cellular networks).

Providing Internet based access is a logical extension of the system. By using web access customers could review the status of current activity or review historical information.

Web access would be provided through a secured web site in order to retain security over the information. Web access would be "view only". Customers would require a username and password in order to access the secure web site.

It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

CLAIMS:

1. A centralised server, or software to program a centralised server, for a mobile application service system, where the server is programmed to create and maintain separate databases for respective clients to hold client knowledge in the form of:

free form database objects, known as frame objects, which hold client data in data fields and are replicated in client computers; and

component based application modules, known as agent objects, which the server despatches to respective clients, as required, to create or manipulate information in the frame objects;

and where the server is programmed such that, after a frame object has been changed a new edition of the frame object, known as a delta frame object, is created which contains only the data fields that have been changed; and synchronisation of the server and the client involves the exchange of edition information between them so that a determination can be made of which delta frame objects need to be aggregated and transmitted in order to bring them into conformity, then transmission of the determined delta frame objects achieves conformity.

2. A centralised server, or software to program a centralised server, according to claim 1, where the client is mobile.

3. A centralised server, or software to program a centralised server, according to claim 1, where objects are held at the server and client in persistent storage.

4. A centralised server, or software to program a centralised server, according to claim 1, where the objects are platform independent.

5. A centralised server, or software to program a centralised server, according to claim 1, where frame objects have data fields added, changed or deleted.

6. A centralised server, or software to program a centralised server, according to claim 1, where conventional compression schemes are employed to improve the efficiency of data transfer.

7. A centralised server, or software to program a centralised server, according to claim 1, where programming and administration tools are provided for the system.

8. A method of providing a mobile application service system, comprising the steps of:

creating and maintaining at a server, separate databases for respective remote clients to hold client knowledge in the form of:

free form database objects, known as frame objects, which hold client data in data fields and are replicated in client computers; and

component based application modules, known as agent objects, which the server despatches to respective clients, as required, to create or manipulate information in the frame objects;

and where, after a frame object has been changed, creating a new edition of the frame object, known as a delta frame object, which contains only the data fields that have been changed; and synchronising the server and the client by exchanging edition information between them, determining from the exchange which delta frame objects need to be aggregated, aggregating those delta frame objects, and transmitting the aggregated delta frame objects in order to bring them into conformity.

9. A method according to claim 8, where the client is mobile.

10. A method according to claim 8, where objects are held at the server and client in persistent storage.

11. A method according to claim 8, where the objects are platform independent.

12. A method according to claim 8, where frame objects have data fields added, changed or deleted.

13. A method according to claim 8, where conventional compression schemes are employed to improve the efficiency of data transfer.

14. A method according to claim 8, where programming and administration tools are provided for the system.

15. A client runtime environment, or software to program a runtime environment, where the runtime environment is programmed to hold: free form database objects, frame objects, which hold client data in data fields and are replicated in a remote server, and component based application modules, known as agent objects, received from the server despatches, as required, to create or manipulate information in the frame objects;

and where the runtime environment is programmed such that, after a frame object has been changed a new edition of the frame object, a delta frame object, is created which contains only the data fields that have been changed; and synchronisation of the client with the server or another client involves the exchange of edition information between them so that a determination can be made of which delta frame objects need to be aggregated and transmitted in order to bring them into conformity, then transmission of the determined delta frame objects achieves conformity.

16. A client runtime environment, or software to program a runtime environment according to claim 15, where the client is mobile.

17. A client runtime environment, or software to program a runtime environment according to claim 15, where objects are held at the server and client in persistent storage.

18. A client runtime environment, or software to program a runtime environment according to claim 15, where the objects are platform independent.

19. A client runtime environment, or software to program a runtime environment according to claim 15, where frame objects have data fields added, changed or deleted.

20. A client runtime environment, or software to program a runtime environment according to claim 15, where conventional compression schemes are employed to improve the efficiency of data transfer.

21. A client runtime environment, or software to program a runtime environment according to claim 15, where programming and administration tools are provided for the system.

22. A method of providing a runtime environment, comprising the steps of:

programming the runtime environment to hold: free form database objects, frame objects, which hold client data in data fields and are replicated in a remote server, and component based application modules, agent objects, received from the server despatches, as required, to create or manipulate information in the frame objects;

and where, after a frame object has been changed, creating a new edition of the frame object, known as a delta frame object, which contains only the data fields that have been changed; and synchronising the client with the server or another client by exchanging edition information between them, determining from the exchange which delta frame objects need to be aggregated, aggregating those delta frame objects, and transmitting the aggregated delta frame objects in order to bring them into conformity.

23. A method of providing a runtime environment according to claim 22, where the client is mobile.

24. A method according to claim 22, where objects are held at the server and client in persistent storage.

25. A method according to claim 22, where the objects are platform independent.

26. A method according to claim 22, where frame objects have data fields added, changed or deleted.

27. A method according to claim 22, where conventional compression schemes are employed to improve the efficiency of data transfer.

28. A method according to claim 22, where programming and administration tools are provided for the system.

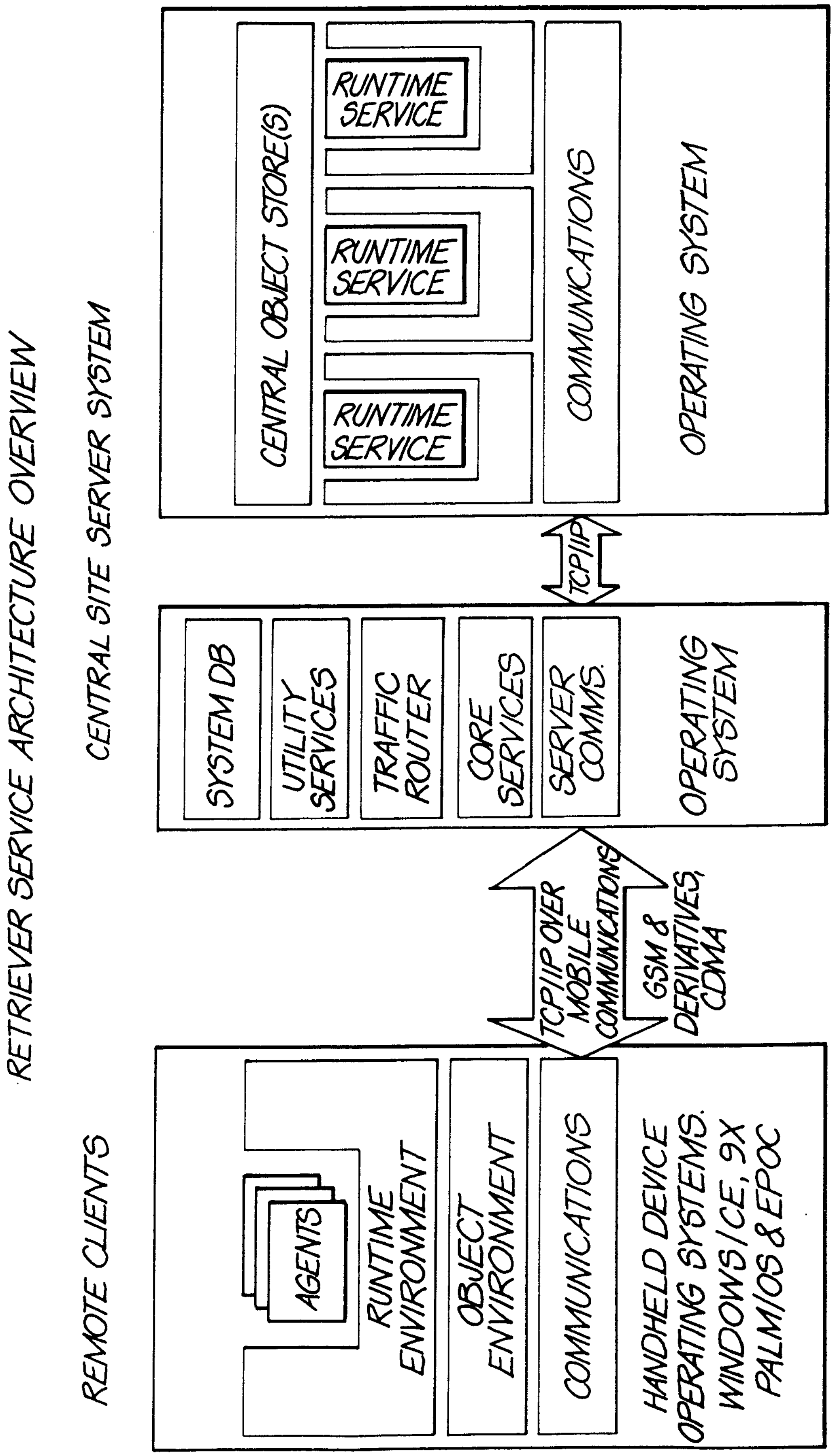

RETRIEVER SERVICE ARCHITECTURE OVERVIEW
REMOTE CLIENTS
CENTRAL SITE SERVER SYSTEM
AGENTS
RUNTIME ENVIRONMENT
OBJECT ENVIRONMENT
COMMUNICATIONS
HANDHELD DEVICE OPERATING SYSTEMS. WINDOWS/CE, 9X PALM/OS & EPOC
TCP/IP OVER MOBILE COMMUNICATIONS
GSM & DERIVATIVES, CDMA
SYSTEM DB
UTILITY SERVICES
TRAFFIC ROUTER
CORE SERVICES
SERVER COMMS.
OPERATING SYSTEM
TCP/IP
CENTRAL OBJECT STORE(S)
RUNTIME SERVICE
RUNTIME SERVICE
RUNTIME SERVICE
COMMUNICATIONS
OPERATING SYSTEM

FIG. 1

RETRIEVER SERVICES ARCHITECHTURE

CENTRAL SITE SERVICE SYSTEM

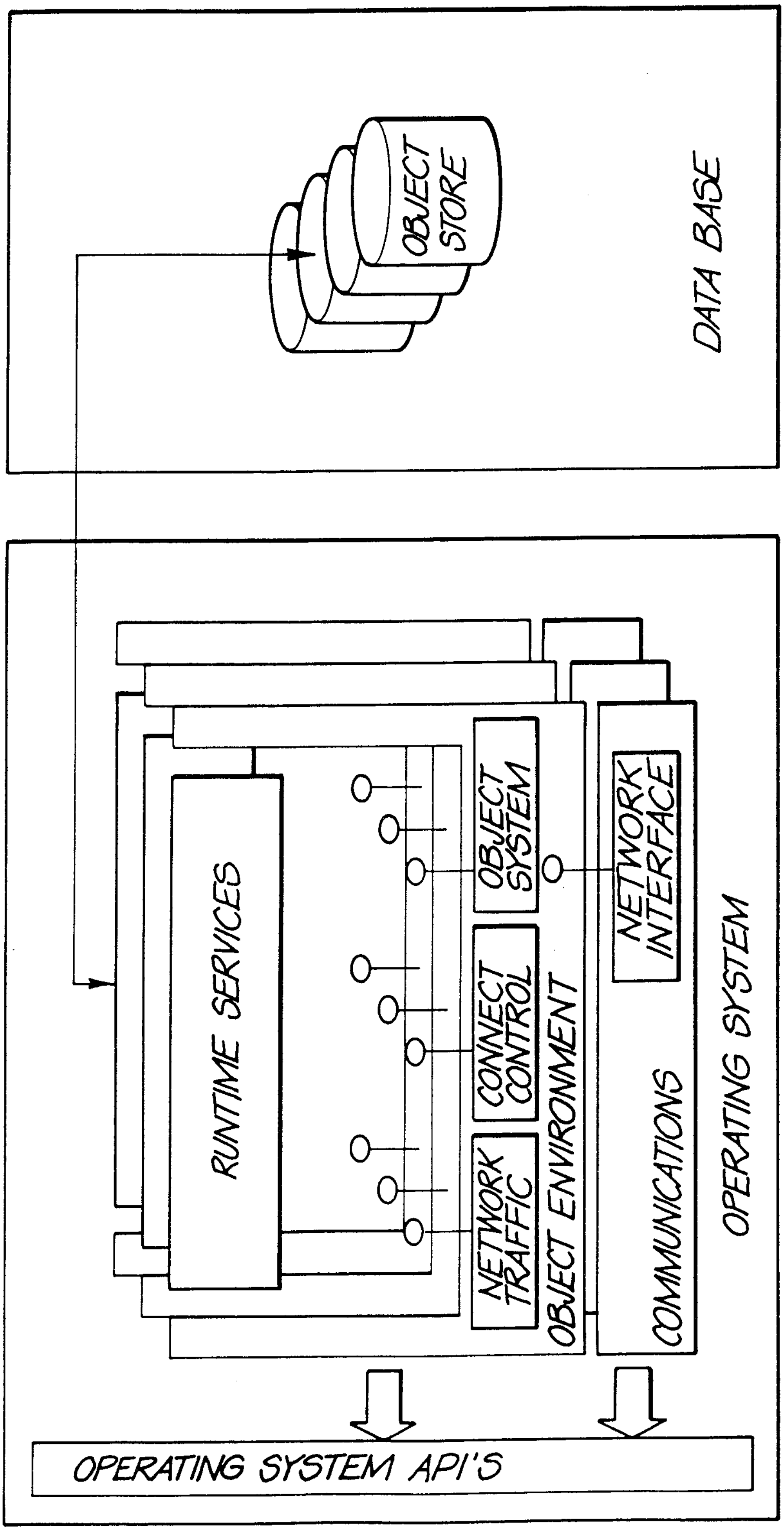


FIG. 2

CLIENT ARCHITECTURE

REMOTE CLIENT

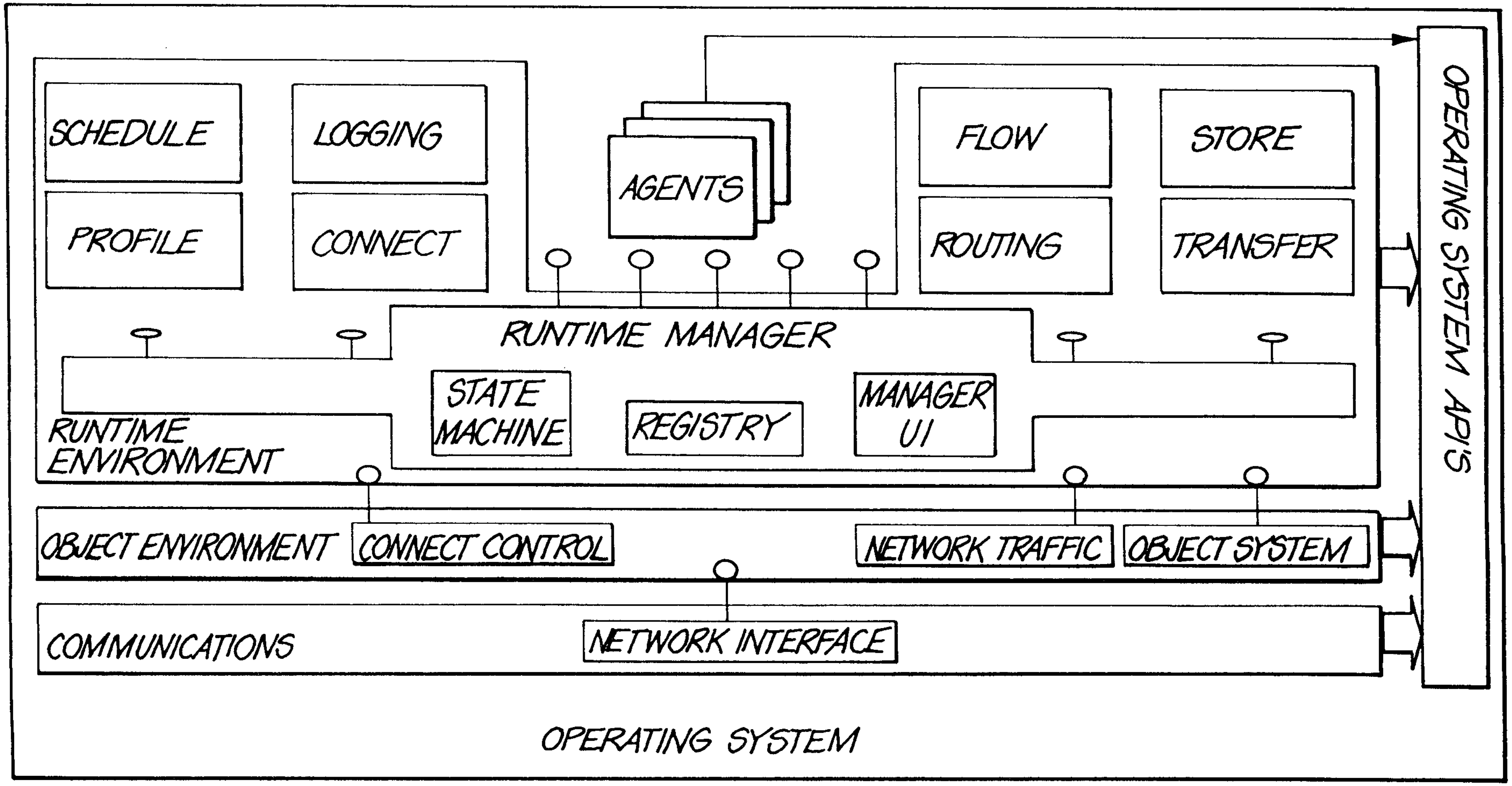


FIG. 3

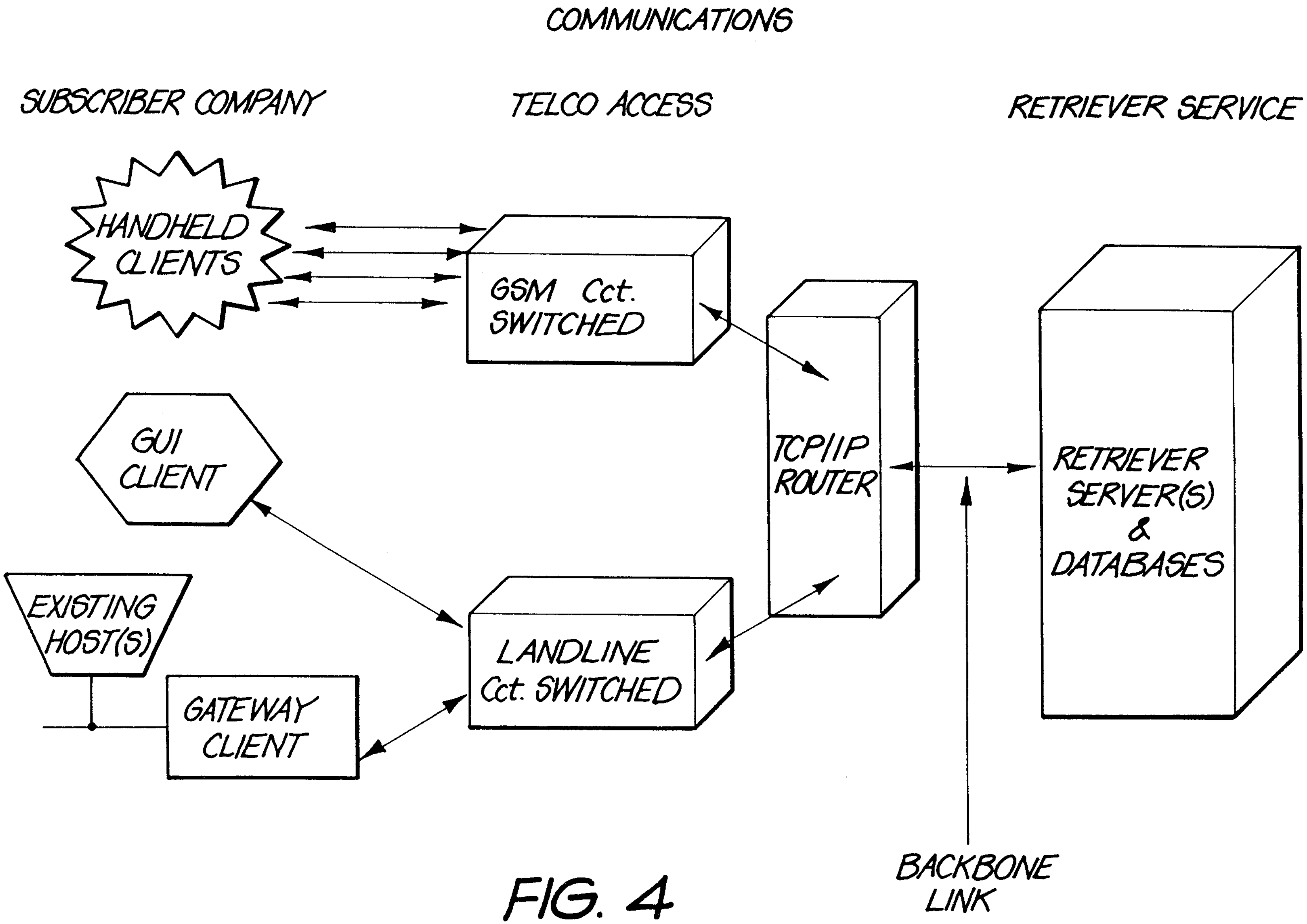

COMMUNICATIONS
SUBSCRIBER COMPANY
TELCO ACCESS
RETRIEVER SERVICE
HANDHELD CLIENTS
GSM Cct. SWITCHED
TCP/IP ROUTER
RETRIEVER SERVER(S) & DATABASES
GUI CLIENT
EXISTING HOST(S)
GATEWAY CLIENT
LANDLINE Cct. SWITCHED
BACKBONE LINK

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU00/00174

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. 7: H04Q 7/24, H04L 12/24, G06F 17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 : AS ABOVE

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPAT, INSPEC (mobile , applications, objects)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	"Sync: A Java Framework for Mobile Collaborative Applications" (Munson et al.) IEEE Computer June 1997 pp 59-66 Whole document	1,2,8,9,15,16,22,23
Y	"Java- and CORBA-based Network Management" (Leppinen et al.) IEEE Computer June 1997 , pp 83-87 Whole document	1,2,8,9,15,16,22,23
Y	"The Mobiware Toolkit: Programmable Support for Adaptive Mobile Networking" (Angin et al.) IEEE Personal Communications August 1998 pp 32-43, Whole document	1,2,8,9,15,16,22,23

☒ Further documents are listed in the continuation of Box C ☒ See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 31 March 2000	Date of mailing of the international search report 19 APR 2000
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. (02) 6285 3929	Authorized officer **DALE E. SIVER** Telephone No : (02) 6283 2196

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
T,A	US 5 913 164 (Pawa et al.) 15 June 1999 Fig. 23A,24,30 column 38 ,line 36 to column 40, line 50	
A	US 5 825 759 (Liu) 20 October 1998 Abstract,figures,claims	
A	"Service Trading in Mobile Environments" (Tokmakoff et al.) ICICS '97 9-12 September pp 417-421 , Abstract,figures	
L	"teach yourself JAVA 1.1 Programming in 24 hours" SAMS ISBN 1-57521-270-6 Copyright 1997 especially page 154 Figure 12.1 Included to establish that a "frame object" is common general knowledge.	
L	"Java Developer's Guide" Copyright 1996 by Jamie Jaworski Section 19.1 Included to establish that a "frame object" is common general knowledge.	

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/AU00/00174

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member					
US	5825759	AU	38189/95	CN	1170493	EP	788720
		JP	10513016	US	5572221	WO	9613951
		AU	68398/96	CA	2227556	CN	1196153
		WO	9704611	FI	971778	US	5585759
		EP	840986				

END OF ANNEX