

# An Architecture for a Distributed 3G Mobile Terminal

Andreas Constantinou, Paul Amery, and Paul Reynolds  
Orange Personal Communications Services Ltd, Bradley Stoke, Bristol, UK

## Abstract

The paper develops an abstract architectural view of a distributed mobile terminal for use in third generation mobile systems, including UMTS and the Mobile Internet. The actual physical form that a third generation terminal will take is still arguable, however, it is possible to be architecturally prescriptive as to the logical form such a terminal should take. The paper starts by reviewing the current status of mobile terminal development and an analysis of the perception of what comprises a third generation mobile terminal. From this analysis hints as to what elements that comprise the architecture are extrapolated. Building upon these findings, the paper presents, for discussion, a conceptual architecture for a 3G mobile terminal, which will impose no restrictions upon implementation. The general operation of the functions that comprise the architecture will be outlined.

The content of the paper is based upon ongoing research work at Orange Personal Communications Services.

## Background

The mobile handset is the most tangible manifestation of an operator's persona, as such obtaining its correct form is vital to the future success of 3G networks. It is the authors' belief that this is best achieved by the acceptance by all players in the 3G value-chain of an open and distributed terminal reference-architecture. Such an architecture would not prescribe the actual form of the 3G terminal but rather allow both openness of interfaces between, and distribution of, functionality.

Before exploring the benefits and architecture of a distributed terminal, it is helpful to outline the customer, manufacturer and operator perspective and particular challenges with the handset architecture being currently deployed. This architecture may also be called the "converged" terminal, indicating that all control and intelligence is converged in a single device.

### User Perspective

- No single device can be expected to provide the perfect solution for individual user requirements. The traditionally niche market of special needs is in fact becoming the mass market. When driving a car for example, a handset user is visually impaired.
- Much of the contained functionality may be redundant from time of purchase.
- The device may be expensive to purchase and difficult to upgrade.
- There is no single device which allows seamless service access. Each physical scenario (e.g. office, car, home) currently requires a separate access device.

### Operator Perspective

- Each time a user has to make a purchasing decision regarding a new terminal, it increases the opportunity for churn (moving to a new operator).
- The costs associated with subsidisation of a handset may mean that no actual revenue is generated for the first 6-9 months.
- In order to enter new markets, such as Automotive/Telematics there needs to be an open, industry standardised handset architecture which the Operator can build upon.
- The customer at present emotionally associates with the handset manufacturer rather than the operator. The loyalty to operator brand is therefore diminished.
- European Operators fear that their ability to influence handset requirements for testing, interoperability and customer needs is decreasing.

### Manufacturer Perspective

- Devices are becoming increasingly expensive to develop, leading to lower volumes and profit margins.
- There is a high cost of associated support required; the more complex the handset – the more likely the need for support.

The mobile terminal marketplace is therefore diversifying as no single device can provide the perfect solution for any user. As this landscape is filled with devices of varying form factors, capabilities and uses, so the need for these devices to interoperate and communicate intensifies. Further, as mobile terminals invade the personal and home space, these devices will become more

successful by integrating in a seamless, intuitive and unobtrusive manner.

In parallel, mobile services are evolving to complement and enhance the users' lifestyle and communication needs, in several areas:

- easier communication: e.g. making it easy to type e-mails, etc.
- greater transparency of the complexity of service creation & management.
- more simple, intuitive, and supportable platform/device and operating system (OS).
- intuitive support of security which becomes more essential, as the distributed terminal becomes more central to the life of the consumer

We may therefore identify two trends within the approaching 3G mobile terminal market. First is the trend for distribution of devices to different form factors, capabilities and intended uses, while being composed of functionally similar components. Secondly, the opportunity arising in this expanding and competitive market is creating the need to maintain a set of common, open standards and interfaces which bind together the terminal components.

Therefore, a 3G terminal architecture must be characterised by the *distribution* of functionality and *openness* of interfaces. Distribution can be thought of as a terminal that has been broken down into its major component parts where each individual part is self-sufficient in support of its functionality. A distributed terminal should support all standard mechanisms for service publishing, discovery, and ad hoc set-up of communications links. Openness can be thought of as the enabler to succeed in the fiercely competitive 3G market; any terminal should seek to standardise on a common target environment for developing applications at the specification and design level. This should ideally allow a write-once-run-everywhere model, so that an application can be delivered and installed on multiple platforms and multiple terminals.

In its simplest form the distributed terminal would consist of a radio module, a keypad an earpiece and microphone; each of these devices connected to the other using, for example, a wireless network. In its most complex form, it consists of a number of entities each with a specific characteristic that may be utilised by the user. A scenario can be visualised where a user seamlessly transfers from a voice call to a video call just by moving in range of their television set.

The user interface for these distributed devices has to be clear and intuitive before the average user will be able to understand how their personal devices are communicating. Since the users cannot be assumed to be computer literate, the devices must be capable of forming simple relationships and clearly indicating these to the user.

### **The Proposed Architecture**

From the above discussion it follows that the key features of a distributed 3G mobile terminal architecture are that it:

- supports delivery to multiple platforms by allowing system components to be built on top of different underlying technologies.
- provides a way of specifying a common target environment for applications at the specification and design level
- separates the interface of the system components from the underlying implementation of those components,
- provides a migration path from proprietary implementations to standards-based implementations.
- allows applications to be designed to a common set of interfaces, and supports rapid deployment of existing applications to new platforms.
- shields the applications from direct access to the underlying platform, providing a model that can be used to implement strict control over the access that applications have to resources on the terminal and the network.

Based upon the above requirements we propose an abstract 6-layer architectural model (see figure 1). At the lowest layer reside two logical entities, the ID module and the terminal hardware. All the upper layers are software-based, starting from the operating system that provides the lowest level access to the hardware. Following that, the execution environment layer abstracts differences in hardware and provides a uniform interface to higher layers. The next layer is a controlled environment for writing services, which provides abstracted access to network-level features and services. The topmost two layers are the standard application and user interface layers. Security is enforced at several layers, from the ID module to the service environment. A significant aspect of the architecture is that, due to the distributed nature of the terminal, the topmost layers need not be physically connected, but may communicate over a thin wireless transport layer.

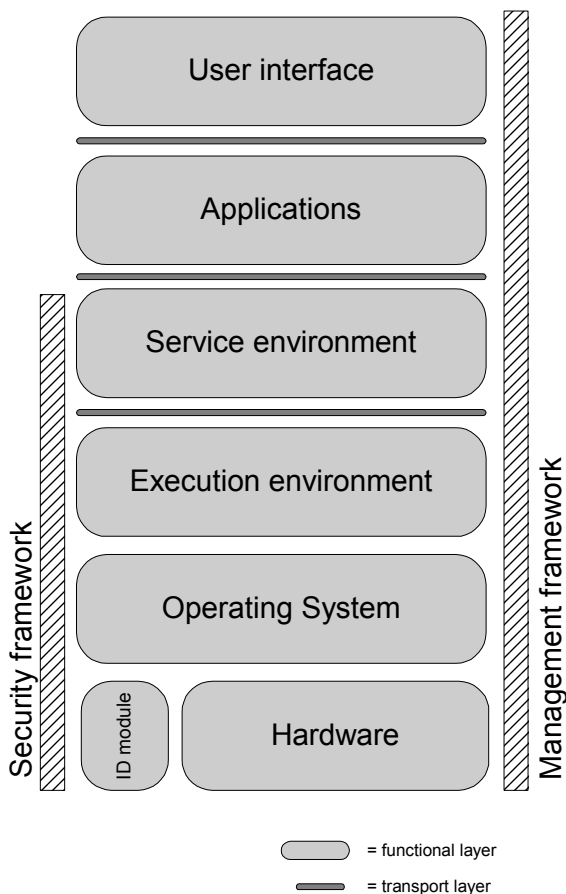


Figure 1: Functional Architecture for Distributed 3G Terminal

### Hardware/ID Module Layer

The layer is comprised of two logical entities, the terminal hardware and the ID module. The former is made up of standard processing, radio, and optionally display hardware, components that are provided by the terminal manufacturer. The ID module is the item that identifies the terminal hardware to the network. In the case where the radio hardware can connect to a wide area network, the ID module also needs to convey user authentication and personalisation information, so that the user may be uniquely identified and the device adapted to the user's settings.

One of the most important parts of the distributed terminal architecture is the radio hardware, which connects the terminal to the either local or wide area networks (LAN/WAN). Popular choices for the network interconnect fabric in the local area are bluetooth and wireless LAN (e.g. 802.11b) technologies. Bluetooth, in particular, is well suited as a terminal interconnect technology, as it is optimised for short range (<10 meters) communications and ad-hoc establishing of point-to-point radio links. In terms of wide area technologies, terminals are moving to incorporate UMTS and GSM radio technology. These

communications can be used to transmit data and audio between the connected devices.

A full set of functions in this layer can be extended to include:

- codec allocation, download and execution
- media conversion (transcoding)
- service brokerage
- flow control
- traffic negotiation & priority
- traffic preemption
- load sharing & resource management
- protocol conversion
- time stamping (windowing)
- routing to multiple terminals & buffering

### Operating System Layer

The operating system (OS) provides a set of application programming interfaces (APIs) to the underlying hardware. Depending on the function and intended use for the terminal, the operating system can be anything from a low specification embedded OS to a fully-fledged system supporting rich graphics and multi-threading. In the case where the terminal acts as an interface between a PAN and WAN, the operating system must support the proxy, router and firewall functionality. The 3G terminal may need to behave as a proxy, or relay, for the other parts of the PAN, or as a firewall providing the routing of traffic from the PAN to the WAN and back to the correct device. The main WAN device has to be capable of signalling the service capabilities of the PAN to the core network. In this way when the user only takes the bare minimum components of the terminal with them, the network knows not to try and set-up, say, a video call to the user. In addition, if the user had not previously been in a position to take video-calls and then they purchase a video capable device they should then be able to receive video-calls.

Within the Operating System Layer it will be necessary to provide for the software to support the management of transport quality and mobility. Transport quality is effected by two mechanisms, material radio losses and hand-offs. Due to the variability of the radio channel, radio losses might occur when the user either moves away from the access point and the strength of the signal decreases or more users are in the vicinity of the user and the opportunity of interference's increases. The access network typically solves radio loss control. The management mechanism has two tools available: forward error code and automatic repeat request. Interference directly affects the terminals ability to communicate, by

knowing the number of users connected to an access network, the terminal can plan the hand-offs to avoid frequency saturation and limit the interference level.

To reduce total cost of ownership, the OS must also be upgradeable 'on the fly' and incorporate remote management functionality. In addition, security must be enforced in terms of authorisation for low level access to hardware.

The most noteworthy operating systems in current use in embedded terminals are Symbian's EPOC, Palm's PalmOS and Microsoft's CE/Pocket-PC family of operating systems. The distributed terminal does not need to rely on a complex operating system, each individual component of the terminal can be as proprietary or as open as necessary as long as each individual component of the is able to interact with the other components of the terminal.

### **Execution Environment Layer**

The execution environment is a software layer that simplifies application development, provides a robust execution environment and enforces a security framework. Application development is simplified by hiding operating system APIs behind a hierarchy of libraries and their exposed objects, as well as providing a unified type model, and version control. In addition, by implementing automatic lifetime management, exception handling, type safety, and just-in-time compilation, the software layer provides a robust execution environment. Finally, by enforcing security, author-based access and execution authorisation throughout the framework, the terminal is protected from unintended use. It should be noted that not all the feature set need be accessible at the execution environment level, but made available as a series of profiles, from which the most profile appropriate for the hardware and intended use is selected. The two most noteworthy approaches to execution environments for mobile terminals are the Sun Java 2 Mobile Edition (J2ME) profiles and the Microsoft.Net Compact Framework.

A further development of the execution environment is the ability to separate those functions that are operator controlled, i.e. a trusted environment, and those functions that are user controlled, i.e. an untrusted environment. The trusted execution environment will process for example, terminal capabilities, tone/message generation, eCommerce accounting and SMS/voice messaging. The untrusted execution

environment will allow the free range of use by the terminal user. Similar boundaries of trust may also apply to other players within the mobile terminal market, e.g. OS manufacturer and device manufacturer. Fully featured terminals should also support memory and process boundaries (also known as sandboxing).

### **Service Environment Layer**

Services are widely recognised as a key to the proliferation of the 3G mobile terminals. For such services to be uniformly accessible, a service environment is required which provides a high level, abstracted and uniform access to all network services. At the same time this environment enforces security over how and when these services are accessed, in accordance with network operator specifications and service plans.

A feature of this layer is that it will 'house' a local database function for storing profile data. For example, to provide for an effective interference reduction hand-off, the terminal needs to store user profiles. Essentially, a user will pay for a certain service and thus it will be allowed to access a number of access networks (e.g. GSM, 802.11b, UMTS).

A key application to be contained within the service layer is support for mobility management. Whilst the mobility management function may be distributed and may reside in both the network and terminal, access technology hand-off decisions will only be made by the network; i.e. any mobility management functionality within the terminal will only provide information to assist that functionality within the network to enhance its performance and efficiency; the network shall always maintain complete control over commanding hand-offs between access technologies. In addition, the location manager should allow for the determination of location based upon:

- Calling Line Identification (CID)
- Global Positioning System (GPS)
- Network generated position record (e.g. triangulation)
- Manual position setting (e.g. user input)

Other data necessary to store includes:

- terminal characteristics
- roaming (preferred networks)
- Quality of Service (QoS) tickets
- tone/message code
- billing, charging and accounting information
- password buffering

## **Applications Layer**

The applications layer of the architecture represents the user applications and shared libraries on the terminal, which are written by third parties. These are built in top of the system components provided in the service and execution layers and make use of the underlying services. It is possible that the implementation of a particular system component resides outside of the platform, either on the ID module or within the network itself. In this case the layer effectively acts as a proxy to the remote application.

## **The User Interface Layer**

The User Interface Layer consists of the tangible interfaces available at the user level. These comprise of a visual display and data entry medium, optional earpiece, microphone, and biometric security interface. The user interface houses a:

- user Client responsible for Session Management & User profiling and management
- emergency Service Client to support of call set-up without authentication providing location, QoS (pre-emption), terminal ID
- user interface function supporting, graphical interaction comprising a display handling, biometric input (e.g. voice recognition, key stroke analysis, etc.) and a user environment (e.g. Virtual Home Environment)

## **The Security Framework Layer**

The security framework should support authentication, authorisation, and basic public key infrastructure services such as encryption, non-repudiation, key distribution and management process for mobility, including:

- biometric analysis sub-system
- digital signature access control
- watermark detection

## **The Management Framework Layer**

The management framework should be capable of supporting a thin client through to a full application-based distributed terminal enabling the dynamic extension of device functionality, including:

- device reconfiguration & tracking
- application and service reconfiguration
- fault diagnosis and solution identification (including remote diagnosis and resolution)
- virus control and detection.

## **Use Case**

An example of a use case for the distributed terminal is given next. A user purchases a low

cost distributed terminal, which is then registered by the operator. Upon entering the local shopping mall, the user decides to take advantage of the large screen pay-per-use video booth, to extend the functionality of the built-in video camera on the handset.

The video booth and distributed terminal instantly associate with each other when in bluetooth range. The user then specifically grants permission for the video booth to access their handset-stored buddy list, which is presented on the large screen video kiosk. A friend is selected to make a video call to. The video kiosk performs video compression and transmits the streaming data to the handset, which has already established a mobile network call upon instruction from the video kiosk with the 3<sup>rd</sup> party, currently at home using a PC. Finally, the session ends, and the user is charged a fee by the video kiosk, via billing and authentication agreement between video kiosk and handset. Note that at each point of the session, the user is asked to grant permission.

## **Conclusions**

The authors believe that by the adoption of an open and distributed terminal architecture, players in the value chain will be able to construct the right solution to meet individual user requirements. It would give the customer a much wider range of choice when coming to purchase the solution that they want. Indeed, it is conceivable that the distributed terminal be supplied by a number of players in the value chain, for example, the radio module could be supplied by a mobile operator, the PDA by computer peripheral manufacture and the earpiece by a haute couture fashion house. Thus a much wider range of manufacturers would be able to provide devices in the personal area network (PAN) space from smart-clothes manufacturers to the more traditional device manufacturers. At the same time these manufacturers would be able to retain value and differentiation, as the market expands to cover different applications, uses and form factors of mobile terminals ranging from high-end devices, e.g. the distributed home, through to minute personal identifiers, e.g. java shirt button.

## **Acknowledgements**

The authors would like to thank their Orange colleagues, Gareth Abel, Matt Bacon, Julie Bale, Peter Brown, Elisabetta Camilieri, Cliff Dennit, Alex Hum, David Townson, Olly Topley, Andrew Thomas and Pierre-Jules Zing-Tsala for their contribution to this paper.