Linux-based 3G Specification

Multimedia Mobile Phone API

Reference Architecture

Document: CELF_MPP_RA_ FR1_20060301

WARNING : This is a working draft for review only, it is NOT a published specification of the CE Linux Forum. It is likely that further substantial changes will be made in the course of review and issue resolution. Send comments on this version to:

MppApiComments@tree.celinuxforum.org

Revision History

Revision	Comment	Reviewer	Editor	Date
1.0	Initial draft for discussion at San Francisco face-to-face		NEC, Panasonic	July 2005
1.0	Minor revisions made at the San Francisco meeting		NEC, Panasonic	July 2005
2.0	Revision for Kawasaki meeting;		Scott Preece	September 2005
2.2.1	Changed to new format		Scott Preece	2005.10.21
2.2.2	Work with figures		Scott Preece	2005.10.24
2.2.3	Revisions based on Tamagawa discussions and San Francisco comments, including NEC comments. Resolution of issues raised previously and first work on harmonization with LiPS Reference Model.		Scott Preece	2005.12.22
2.2.4	Revisions to resolve Andre's issues		Scott Preece	2006.1.11
FR1	First version for Formal Review		Scott Preece	2006.3.1

75

0. Introduction

This document describes a reference architecture of Linux based 3G multimedia mobile-phone, developed by the CE Linux Forum's Mobile Phone Profile Working Group. A "reference architecture" is an abstract model for a class of software systems that is widely agreed to as the common model for that kind of system.

The Architecture in this document is based on an architecture that was originally the collaborative work NEC Corporation, Panasonic Mobile Communication Ltd., and NTT DoCoMo, Inc.

The basic architecture is described in chapter 1.

The functions of each component of the architecture are described in chapter 2.

The data and control flows between components are described in chapter 3.

0.1 Scope

This document defines the reference architecture of Linux based mobile phone. This is a non-normative part of the Specification. The goal of the Reference Architecture is to provide the context for the descriptions in the normative parts of the Specification.

The Reference Architecture does not describe the internal architecture of the communication protocol stack or the Application Framework, beyond the semantics exposed in the API.

0.2 Vocabulary and Abbreviations

See the corresponding section in the Preface document.

0.3 Reference

[1] GTK+ API Documentation (http://www.gtk.org/api/)

[2] GNOME GTK+ Reference Manual (http://developer.gnome.org/doc/API/gtk/index.html)

[3] GNU C Library (http://www.gnu.org/software/libc/manual/html_mono/libc.html)

[4] X.org http://www.x.org/

1. The Mobile Phone Domain

The WG is defining one or more Reference Architectures - standard organizations of components for mobile phones. The architectures will correspond to one or more Reference Tiers, which are broad classes of phones with common characteristics (and, therefore, suitable for sharing an architecture).

The table below gives working definitions of the Reference Tiers. Note that these represent a particular point in time (and-of 2004) and some of the details of specific areas will change as technology changes. Many products do not fit neatly into one niche and will blend characteristics of different tiers. Also, many products will add features not covered by this table.

	Tier			
Aspect	Smart Phone	Multimedia Phone	Feature Phone	Plain-Old Mobile
Focus	business focus	Personal/Entertainment Focus	Lifestyle Focus (voice plus social networking support features)	Voice
Primary Functionality	Full PDA functionality (Calendaring, address book)	Strong PIM support, personal content management features	Minimal PIM functionality (phonebook, datebook)	Phonebook and call logs
Extensibility	Extensible (downloadable features)	Limited extensibility (MIDlets or BREW)	Limited extensibility (MIDlets/BREW)	No extensibility
Multimedia	Optional	Vido capture support, Media/content players, stereo	Limited multimedia support (pictures, MP3, MIDI,Simple, low-frame-rate animations)	None
DRM	Optional	Multiple DRM schemes	Hard DRM (limits on copying any media of given types)	None
Camera	Optional	2-3 megapixel camera	VGA camera or no camera	No camera
Browser	XHTML Browser	XHTML Browser	WAP Browser (text-centric)	Embedded access to specific URLs
Display	QVGA or larger color display	QSIF or larger color display	QSIF or smaller color display	Small display (64x96), non-

				color
Interaction	Touchscreen UI or QWERTY keyboard plus pointing device	Specialized keypad for media/game interaction	Standard keypad plus carrier-specific keys	Standard keypad
Connectivity	3G connectivity, possibly WLAN, Bluetooth, IrDA	2.5G or 3G connectivity, possibly WLAN; High-speed USB; Bluetooth	2G connectivity; USB or serial cable	2G connectivity; proprietary accessory cable
Memory	32M RAM, 64M ROM, removable storage	64M RAM, 64M ROM, Hard Disk or large removable storage	16M RAM, 16M ROM, no removable storage	8M RAM, 8M ROM or less
Processor	120MHz	200MHz	30MHz	15MHz

108

2. Architecture

This document describes the architecture of a mobile phone based on the Linux operating system. The architecture separates processing between two domains: the application domain and the communications domain. The two domains might run on separate processors, as separate processes on a single architecture, as separate virtual procesors running over a micro-kernel, or other physical implementation. The Communications domain would include all activities requiring hard real-time behaviour.

The Reference Architecture identifies this partitioning of processing domains but does not dictate the physical realization used. In Figure 1, the Bridge represents whatever mechanism is used in the particular realization to support interaction between the domains. The implementation of the Bridge not in the scope of the Reference Architecture.

The blocks shown with a white background in Figure 1 are not part of the scope of the Mobile Phone API. They are shown as part of the common understanding of the structuring of a Linux-based mobile phone.

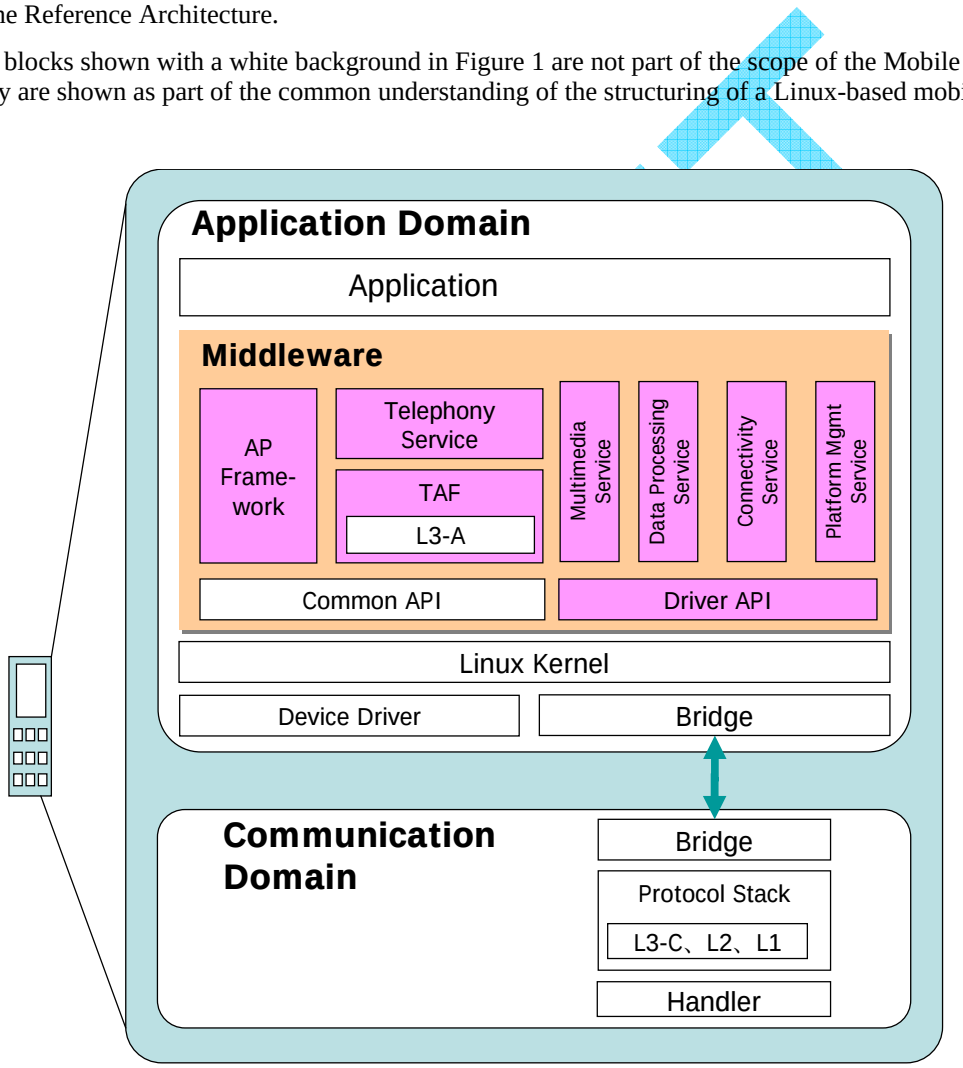


Figure 1 Overall Architecture

2.1 Applications Domain

The software running in the Application Domain contains the following 4 layers:

Application

Middleware

Linux kernel

Driver & Bridge

2.1.1 Application layer

The application layer contains various applications. They are classified into the following 8 categories; samples are listed to illustrate the categorization, but are not meant to be requirements nor to be a complete list of the applications in a given category:

2.1.1.1 Telephony applications

Telephony applications include Standby Screen, main menu, videophone application, phone applications, phonebook, NW service and phone function setup, etc.

2.1.1.2 System applications

System applications include Air download, Generic LCD display, Backside LCD display, PIN authentication and monitor mode, other function setup, Equipment alarm, etc.

2.1.1.3 Multimedia applications

Multimedia applications include still image viewer, video viewer, camera app, vector graphics viewer, avatar and ring tone management, etc.

2.1.1.4 Data-processing applications

Data-processing applications include OCR, barcode, SD-PIM, data transfer, memory transfer, external I/F communication, user data, IR, schedule, voice memo, schedule alarm and data folder, etc.

2.1.1.5 Internet-applications

Internet applications are those that use TCP/IP to access resources on the internet, including e-mail, Browser, HTML mailer, etc.

2.1.1.6 Internet Application Engine

Internet application engine include engines for HTTP, SSL, embedded languages, etc.

2.1.1.7 Java Application Engine

The Java applications engine includes a Java Virtual Machine, JAM, and class libraries.

2.1.1.8 Others

Others includes Accessory menu, Accessories (text memo, calculator etc.), etc..

2.1.2 Middleware layer

Middleware layer contains the following components.

2.1.2.1 Applications framework

The Applications framework provides application developers with a common framework of services commonly used by mobile-phone applications.

2.1.2.2 Telephony service

The telephony service provides application developers with a framework of services for communications and handset management.

2.1.2.3 Multimedia service

The multimedia service provides video phone service (H324, for example), and multimedia decoding, encoding, and rendering facilities.

2.1.2.4 Data processing service

The data-processing service supports processing the data from various devices, e.g., bar-code reader, optical character reader, etc.

2.1.2.5 Platform Management service

The Platform Management Service provides the functions of system management, including installation of software and control of system processes.

2.1.2.6 Connectivity Service

The Connectivity Service handles inter-device functions, such as synchronization and OBEX data exchange.

2.1.2.7 TAF (Terminal Adaptation Function)

The TAF provides access to communication services. It consists of voice communication TAF, packet communication TAF etc.

2.1.2.8 Common API

The Common API provides application developers with standard C-language functions.

2.1.2.9 Driver API

The device-driver API provides middleware and application programs access to devices and to services modeled as devices.

2.1.3 Kernel/Driver layer

2.1.3.1 Kernel and Device Drivers

The Kernel / Driver layer contains the Linux Kernel, device drivers, and the Bridge.

2.1.3.2 Bridge

The Bridge supports communication between the Application and Communication domains, which may be implemented as separate processors or not, but will minimally have different scheduling regimes.

2.2 Communications Domain

The Communications Domain performs all processing that requires hard real-time behaviour, including executing the lower levels of the protocol between the device and the network. Its protocol stack contains the network stack's L1, L2, L3 layers. The Bridge supports communication with the Applications Domain.

3. Description of functional entities

3.1 Linux Kernel

The Linux Kernel provides:

Memory and CPU management

Timer and system clock management

Process management: create, destroy and dispatch

File-systems: files, directories, and space management

Console handling

Inter-process-communication: sockets, message queues, shared memory, etc.

Network communication: TCP/IP

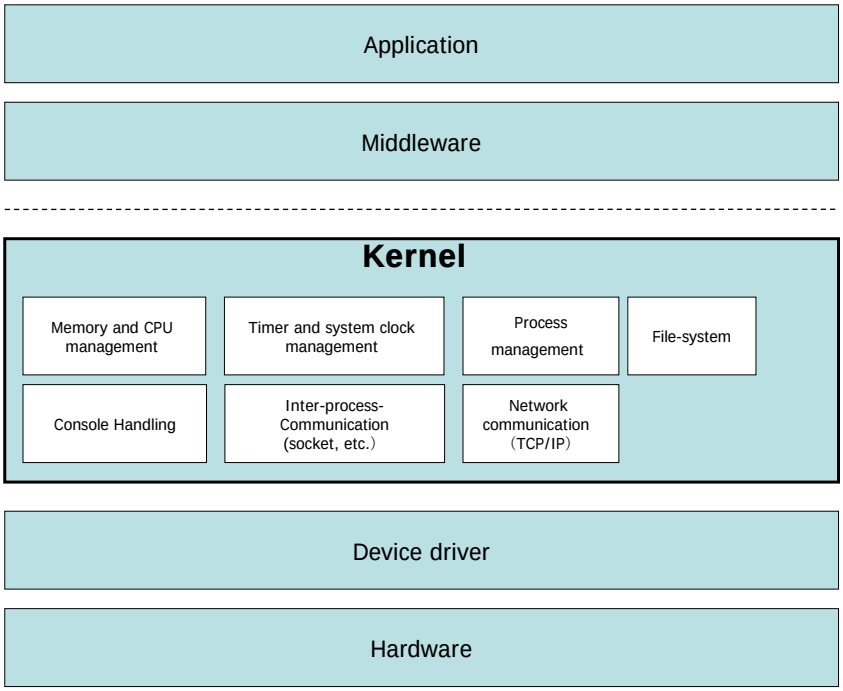


Figure 2 Linux Kernel

3.2 Common API

The Common API contains various functions for applications written in, including the standard C libraries. The Common APIs conform to the POSIX standard.

3.3 AP Framework

Fig-3 is an overview of the application framework. This is an area where variation is common in the domain, with the specific requirements of the application set and the UI framework chosen for specific products potentially differing from this reference architecture, Usually, however, the functional separation

is similar to this.

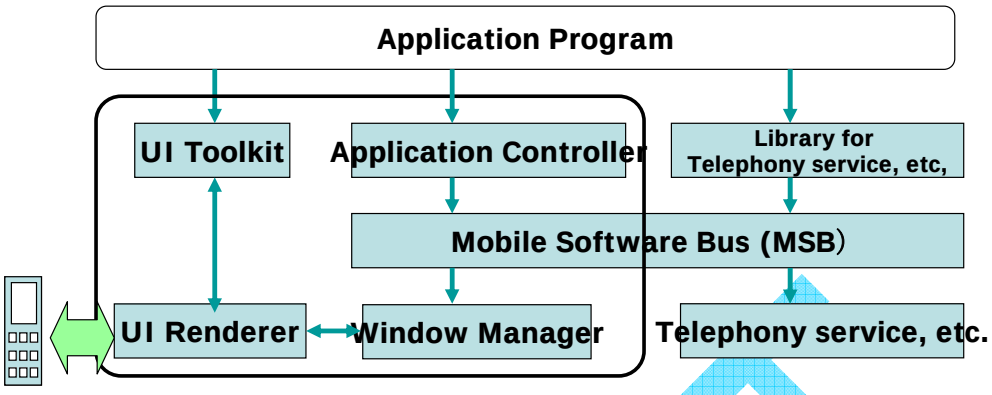


Figure 3 AP Framework

3.3.1 Window Manager

The Window Manager provides unified operation and decoration for windows and controls overlap of windows (clipping and layering).

In the Application framework for mobile-phone, window manager provides:

Foreground and background control of application window

Tracking the state of applications (active, inactive, idle, not-started)

3.3.2 Application Controller (APC)

The APC controls the start and end of applications. That is, starts applications, switches between applications, andshuts down applications at power-off. APC also manages application start status, and controls application switching operation (e.g., selection of an application to be next used as a foreground application at application switching).

The Window Manager controls window stacking, focusing, and properties for each group during the period from window generation to deletion. The Window Manager receives requests from the APC library through the MSB and resolves contention between applications, according to the priority table based on the application status. It also control windows overlapping and key focusing.

The Application Controller uses the MSB to communicate between Application and Window Manager. By using the functions of the window manager application developers do not need to know the functions of MSB.

3.3.3 UI Toolkit

A set of functions and facilities for the application to describe its interaction with the user.

3.3.4 UI Renderer

The UI renderer controls presentation of the interaction elements on the display(s).

3.3.5 Mobile Software Bus (MSB)

The Mobile Software Bus (MSB) supplies communication services (synchronous and asynchronous communication) between applications and services. Different implementations of the reference architecture may use different kinds of inter-process communication for this role.

3.3.6 Others

3.3.6.1 PICT (PICTograph) display library

It controls turning on/off of the upper pictograph elements such as antenna and battery.

3.3.6.2 Image library

It Converts, extracts, or compresses image data, aloes used to set or acquire image information.

3.4 Telephony Services

Fig-4 describes the telephony processing framework. The link shown as “Socket communication” could also be implemented using other means of interprocessor communication, or inter-process communication on a single-processor system.

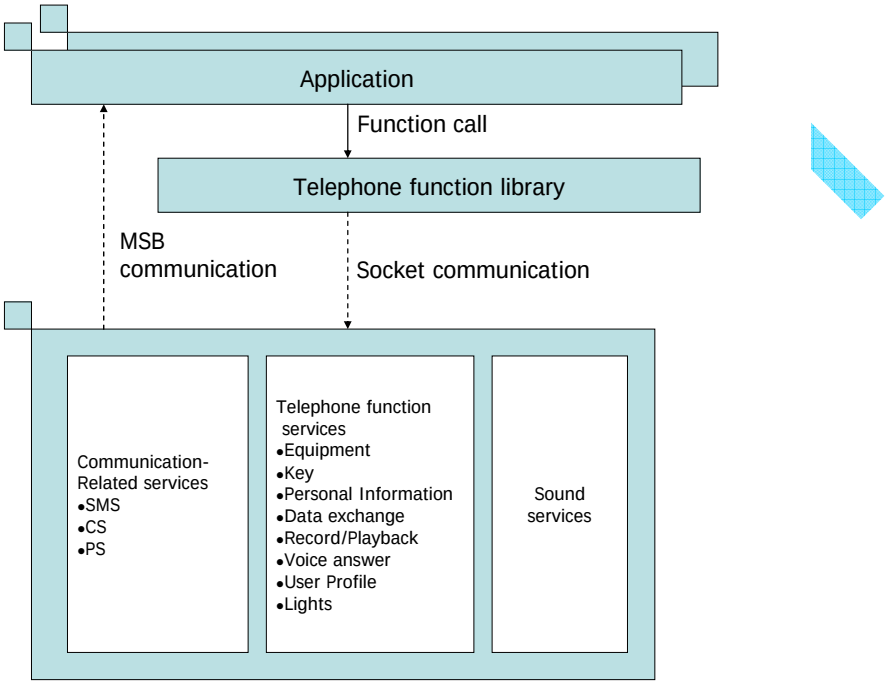


Figure 4 Telephony Processing

3.4.1 Circuit-Switched (Voice) Communication service

It provides application programs with dialing, call disconnection, rejecting incoming call etc. for circuit-switched communication service.

3.4.2 Packet-Switched (Data) Communication service

It provides application programs with initiation, termination, rejecting incoming call etc. for packet-switched communication service.

3.4.3 SMS Communication service

It provides application programs with notification of events and status of SMS service etc.

3.4.4 Equipment service

It provides application programs to setup, control, and read the status of various handset hardware elements (batteries, headsets, etc.).

3.4.5 Personal Information Manager

It provides application programs to register a schedule, sort the schedule data, read to-do data etc.

3.4.6 Data Exchange

It provides application programs with the functions to handle phone-book, memo, image, and video, etc. on internal and removable memory stores.

3.4.7 Record and Playback

It provides application programs with record and playback of voice memo.

3.4.8 Light Management

It provides application programs with functions for controlling various lights.

3.4.9 Sound System

It provides application programs with functions for ring-tone and melody of the equipment.

3.4.10 User Profile Library

It provides application programs with ability to read and set various attributes of the user's profile, such as registered phone numbers, owner name, and e-mail addresses.

3.4.11 Removable Media Management

It provides application programs with the functions for removable media, e.g. read, save, delete etc.

3.5 Multimedia Framework

Fig-5 describes the multimedia framework.

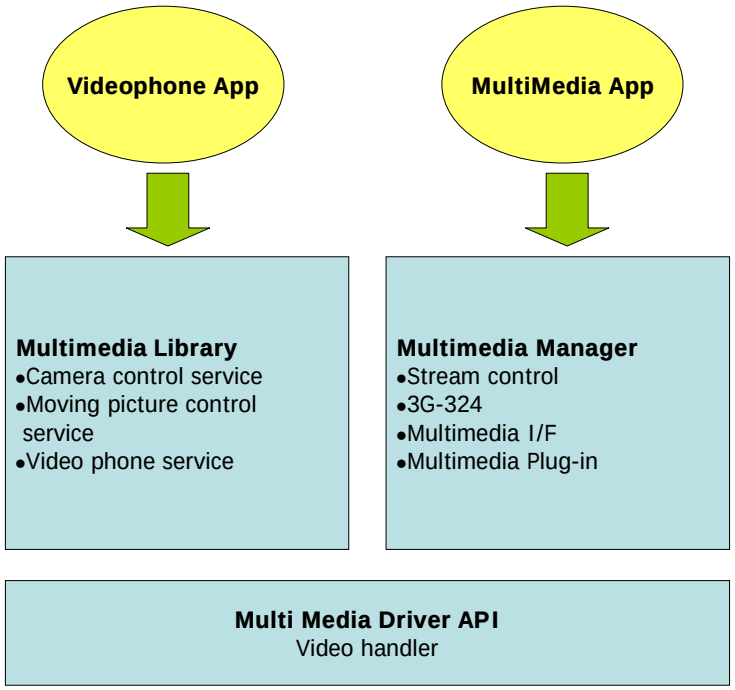


Figure 5 Multimedia Framework

3.5.1 Multimedia Manager

The Multimedia Manager provides interfaces for interconnecting multimedia functions, such as video phone library, 3G-H324M, camera control library, and moving picture control library.

3.5.2 Multimedia Library

The Multimedia Library provides interfaces for camera control, moving picture control, and video phone services.

3.5.3 Multimedia Driver API

Multimedia Driver API provides video handler interfaces.

3.6 Data Processing

This block covers various kinds of add-on functionality related to dealing with external data

3.6.1 Bar Code Library

It provides the bar code reader functions.

3.6.2 OCR Library

It provides the OCR access reader functions.

3.6.3 Location Services

This provides access to the geographical location of the phone and events related to recognizing specific locations, for phones equipped with the appropriate capabilities.

3.7 Connectivity Service

The OBEX (object exchange) module supports synchronization and sharing of information between devices by exchange of data objects. It provides an interface that is used by OBEX to perform communication processing based on request messages from application programs and to return processing results to the application programs. It also provides an interface that is used by OBEX to convert objects to canonical format for interchange.

3.8 Platform Management Service

The Platform Management monitors the activation of each task at the time of power on and the deactivation of each task at the time of power off, as well as the charging and other statuses of the mobile terminal.

3.9 Driver API

The Driver API provides upper layer components (middleware and application programs) an abstract interface to device drivers, so that device-independent components do not need to be aware of the particular device drivers available on a specific handset. This avoids hardware dependencies, enabling development of portable middleware and application programs for mobile phones.

4. Data Flows

This section describes data and control flow between components of the architecture in various domains.

4.1 Voice communication

Telephony application controls C-plane using the telephony service.

Voice data is decoded by a protocol-specific codec and sent to the appropriate audio output.

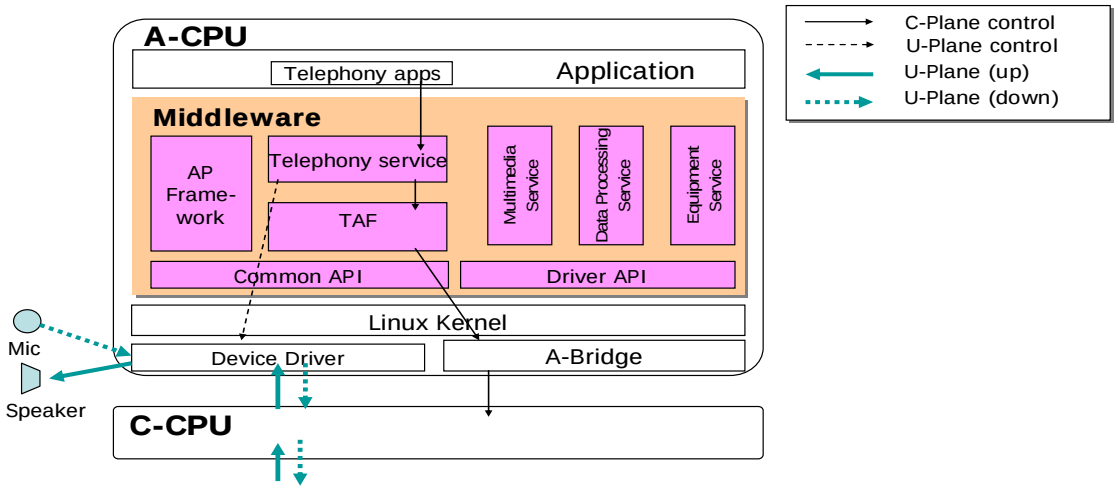


Figure 6 Voice Communication

4.2 Video phone

Video phone application program controls C-Plane by telephony service.

Multimedia service provide decoding and encoding facilities to support video telephony, such as H.324.

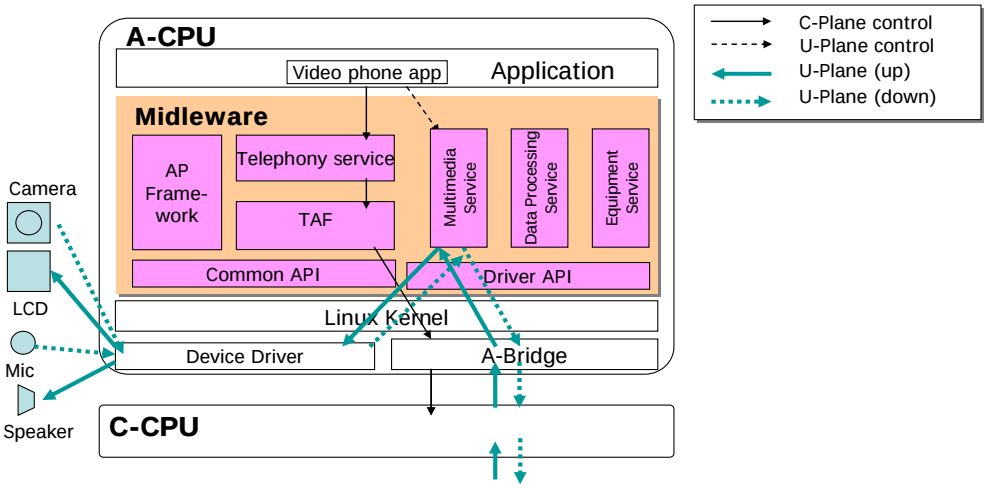


Figure 7 Video Phone

4.3 Internet Application

Internet Application program controls C-Plane by telephony service.

Data from the Internet is transported by a protocol stacked based on TCP/IP using the Linux Kernel networking capabilities.

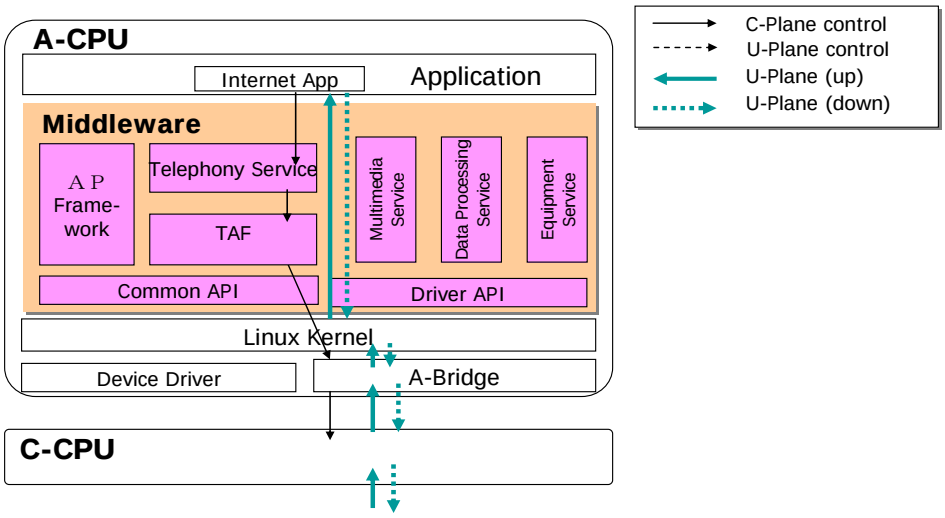
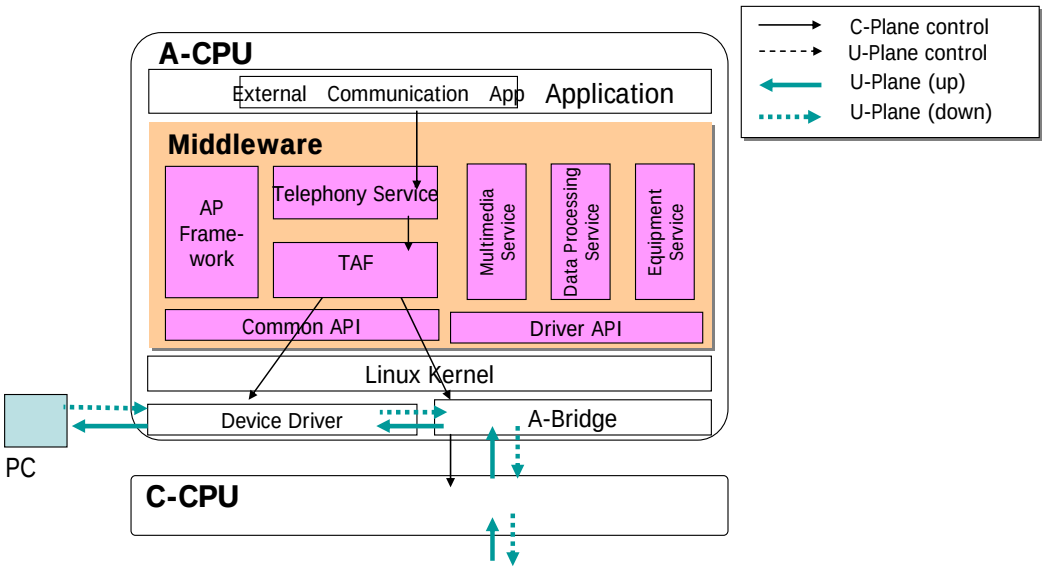


Figure 8 Internet Application

4.4 Dial-up Networking with External Devices

Applications program for dial-up networking controls C-Plane by telephony service.

Data are received and transmitted to an attached external device through USB or other communication bus and device driver and routed back to the internet through the communications stack.



337

338 **Figure 9 PPP Communication**

339 4.5 SMS communication

340 Telephony Service SMS library controls C-Plane and U-Plane to send and receive short messages.

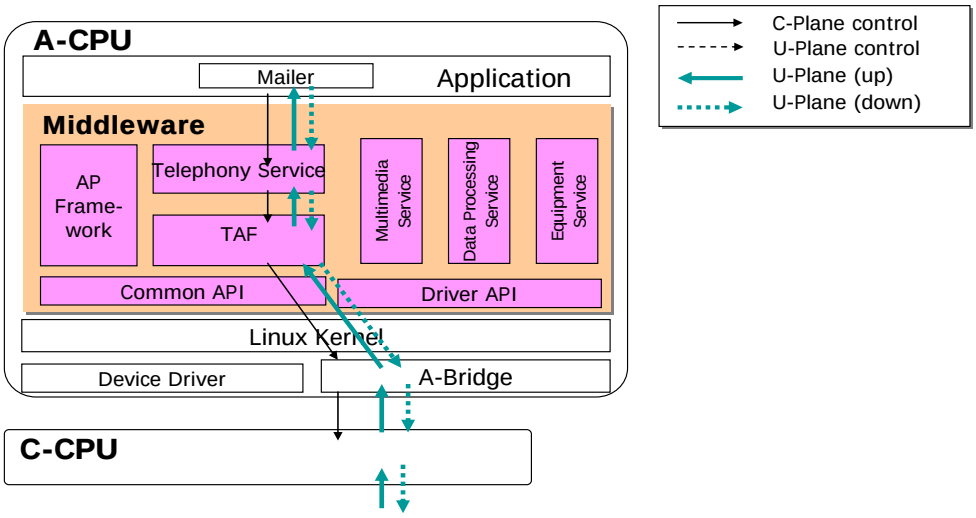


Figure 10 SMS Communication

