OSTI Technology for Open and Secure Mobile Terminals

The OSTI Architecture Specification has been established to specify the terminal architecture for achieving open and secure mobile terminals. This will enable enterprises and system integrators to select the applications that they need and install them in mobile terminals. This research was conducted jointly with Intel Corporation.

Takehiro Nakayama, Ken Ota and Atsushi Takeshita

1. Introduction

NTT DoCoMo together with DoCoMo Communications Laboratories USA and Intel Corporation has established the Open and Secure Terminal Initiative (OSTI) Architecture Specification [1] to satisfy the need for using existing application software on mobile terminals. In the past, when enterprises and individuals developed software such as native applications[*1] for use on mobile terminals, it was difficult to maintain the same reliability and security as traditional mobile terminal services. We have worked to solve this issue by adopting multi-domain architecture that establishes separate environments (domains) for executing software on a mobile terminal. We researched multi-domain architecture in collaboration with Intel Corporation in light of their extensive hardware knowledge.

While technology for multi-domain architecture for servers and personal computers has already reached a mature stage, applying it to mobile terminals requires specific considerations to ensure that phone services are provided without compromise regardless of which domain is currently being used. OSTI provides a mechanism for switching between peripheral devices allocated to each domain (e.g., keyboard, display, speaker, microphone), a mechanism for inter-domain communication, and a mechanism for event-driven interrupts. These mechanisms enable mobile-terminal services to be maintained in a consistent manner.

In this article, we begin by outlining OSTI multi-domain architecture and describing its features. We then outline OSTI specifications for two underlying implementation technologies—OS Switching and Virtual Machine Monitor (VMM)—that can be used for achieving the multi-domain architecture specified by OSTI. Finally, we touch upon future directions for OSTI.

2. Multi-domain Architecture

2.1 Overview of Multi-domain Architecture

As shown in **Figure 1**, OSTI architecture supports two domains in the application unit[*2]: an Enterprise Domain that can be used as desired for existing corporate applications, browsers, and other business applications, and an Operator Domain that provides traditional operator services such as phone calls and mails. Each domain may have a different OS as well as its own Graphical User Interface (GUI) and security policies.

The OSTI specification calls for a Domain Abstraction Layer (DAL) that acts as an interface between the Enterprise Domain and the main terminal platform. It allows for flexibility in implementing DAL and does not mandate a particular implementation technology. Instead, OSTI specifies an abstract interface and presents OS Switching and VMM as two examples of underlying implementation

[*1] **Native application**: Application software that can be directly executed on a general-purpose OS. An i-appli executes on the Java™ Virtual Machine and is therefore not a native application (Java is a trademark of Sun Microsystems, Inc., USA).

[*2] **Application unit**: Many mobile terminals adopt a two-CPU architecture consisting of a Communication CPU (C-CPU) that controls the communication section and an Application CPU (A-CPU) that controls the application section. The application unit refers to the section controlled by the A-CPU.

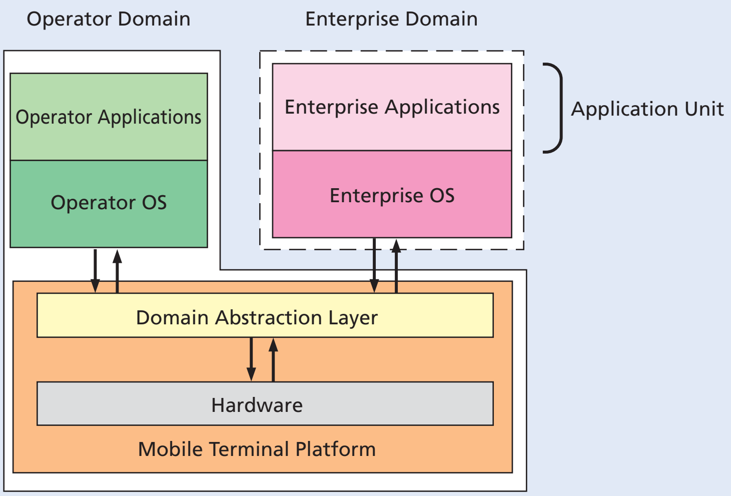

Figure 1 OSTI architecture

technologies. The OS Switching system exploits the suspend and resume functions found in many OSes to switch between two OSes in a mutually exclusive manner. The VMM, on the other hand, provides a set of virtual devices corresponding to the CPU, memory, and peripherals and has two OSes operate in parallel. The aim of DAL in the OSTI specification is to minimize the above differences.

2.2 OSTI Specification Features

1) Guarantee of Secure Operator Services

The most important feature of the OSTI specification is that the Operator Domain is protected from being negatively affected by the Enterprise Domain even if a security issue should arise in the latter. This separation of domains enables the reliability and security of operator services like phone services to be guaranteed as in the past. The Operator Domain is operated under the same operator-established security policies as those for traditional mobile terminals and is consequently provided with the same level of reliability and security. In contrast, the security policies of the Enterprise Domain may be determined by other than the operator such as a corporate IT department. Those policies will dictate which software is allowed to be used in the Enterprise Domain.

2) New User Experience

Another feature of the OSTI specification is that it enables multiple domains to coexist on the same mobile terminal providing the user with a new way of using a mobile terminal. A typical scenario would be to use the Enterprise Domain for business applications and the Operator Domain for personal use. If, for example, Windows®[*3] were to be used as the OS in the Enterprise Domain, the user would have a wide range of business applications to choose from.

Photo 1 shows an external view of an OSTI prototype system implemented by NTT DoCoMo and DoCoMo Communications Laboratories USA. This implementation uses Linux in the Operator Domain and Windows Consumer Electronics (CE) in the Enterprise Domain. In the OSTI specification, the user may interact with only one domain at any one time, and may switch to the other domain by, for example, pressing down on a button dedicated to that function. The domain that the user is currently interacting with is called the "foreground" domain while the domain that the user is not interacting with is called the "background" domain. Whether or not software running in the foreground domain continues to operate after switching that domain to the background depends on the implementation.

3) Consistency in Mobile Terminal Services

Multi-domain architecture must maintain the functionality and usage format of traditional mobile terminals. For example, if domain switching is performed after setting silent mode (manner mode) in the foreground domain so that the other domain now becomes the foreground domain, the silent-mode setting must be reflected in that domain.

The OSTI specification provides basic functions and protocol for exchanging messages between domains to satisfy this requirement for inter-domain linking and cooperation. It specifies, in particular, header elements necessary for exchanging messages such as a unique message identifier, an identifier for distinguishing between ordinary data messages and acknowledgment messages, message length, and position of next message, and

[*3] **Windows®**: Windows is a registered trademark of Microsoft Corporation, USA.

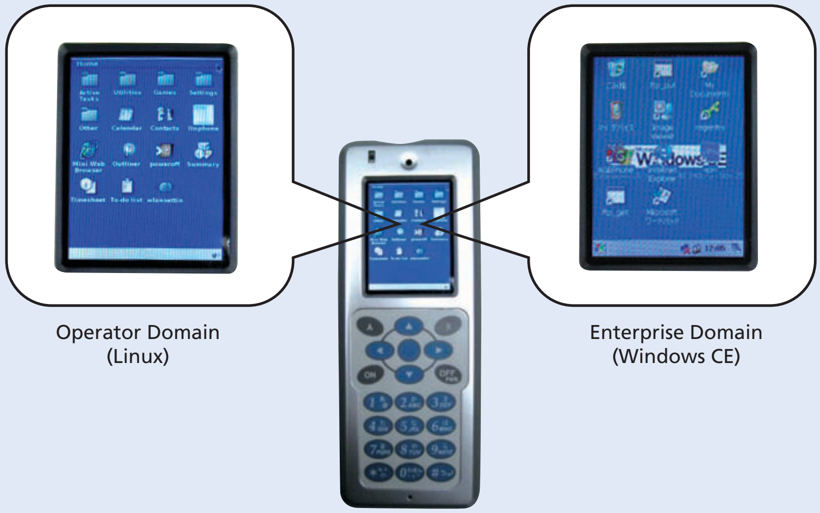

Photo 1 External view of OSTI prototype system

a means of conveying acknowledgment (successful/failed) in message exchanges. This message-exchange function enables the Enterprise Domain and Operator Domain to perform cooperative processing.

3. OS Switching

This chapter describes OS Switching as one method for implementing the OSTI specification. Many OSes today provide suspend and resume functions for entering sleep mode to save power. When an OS enters a sleep state by the suspend function, the system's state is saved in memory and most hardware is turned off. Then, when returning from this sleep state by the resume function, the system's state is recovered from memory and the system is restored. At this time, any suspended applications resume from the point at which their execution was halted. The OS Switching method uses these suspend and resume functions to switch between domains. When switching a foreground domain to the background, the domain makes a transition to the sleep state by the OS suspend function, and when switching a background domain to the foreground, the domain returns from the sleep state by the resume function. This switching operation can be accomplished in a fraction of a second.

OS Switching includes a restriction that software in the background domain does not operate. As a result, processes for playing music, performing backups, etc. using background-domain software cannot be executed while the user is interacting with the foreground domain. There are also requirements concerning network connections. Either all network connections are cut off at the time of domain switching or special processing must be performed to maintain network connections. In the latter case, information related to the connection state must be shared with the background domain by inter-domain communication, and processing must be performed to recover the network-connection state when the background domain returns from the sleep state.

The OSTI specification prescribes the following three items as an interface between the Enterprise Domain and DAL when using OS Switching.

- Enterprise OS launch, pause, and resume
- Enterprise OS system management interface:
  platform management, data communication between Operator OS and Enterprise OS, storage, peripheral devices
- OS Switching specific considerations:
  OS switching control, Operator Domain protection

Among the above items, this article describes OS switching control and Operator Domain protection as special study items for OS Switching.

3.1 OS Switching Control

OS switching between domains may be controlled not only by user interaction but also by a switching operation based on specific event interrupts. For example, if a call is received while the Enterprise Domain is in the foreground state, control could be passed to the Operator Domain. In this case, the Enterprise Domain transmits the incoming-call event to the Operator Domain by inter-domain communication and issues a domain-switching request to the DAL. Then, after resuming, the Operator Domain receives the incoming-call event and executes incoming-call processing. However, as OS Switching is a system in which OSes operate on a mutually exclusive basis, there may be some situations in which disabling of switching is desired such as during a Voice over IP (VoIP)[*4] call by the Enterprise OS. To satisfy this need, the OSTI specification provides an interface for handling OS-switch lock/unlock requests.

3.2 Operator Domain Protection

When the Enterprise Domain in OS Switching architecture is in the foreground, the OS kernel[*5] of that domain can, in principal, access all resources on the mobile terminal. For this reason, resource-access conditions are generally imposed on the Enterprise OS to protect important data storage areas in the Operator Domain (e.g., areas that store the user's address book or decryption keys for using paid content). This assumes that only an Enterprise OS deemed reliable by the operator would be selected for the Enterprise Domain. But if, by some chance, a malicious program were to gain OS kernel privilege, the danger exists that important data in the Operator Domain could be abused. In response to this problem, the OSTI specification offers two candidate mechanisms for protecting data: 1) obfuscation-hardened encryption and integrity protection and 2) TrustZone®[*6]-based protection.

Obfuscation-hardened encryption and integrity protection encrypts important data when the Operator Domain enters sleep state by domain switching. This mitigates the danger of data abuse by the Enterprise Domain while the Operator Domain is in sleep mode. When domain switching is again performed and the Operator Domain returns from a sleep state, the encrypted data is decrypted and restored. But here, if the key for encryption processing is stored as-is in memory, it could be stolen by the Enterprise Domain while the Operator Domain is in sleep mode thereby enabling crucial data to be decrypted. In light of the above, the OSTI specification mandates the use of obfuscation techniques when encrypting data in OS Switching. Such a technique enables a key to be reproduced only by an algorithm that is difficult to reverse engineering[*7].

TrustZone-based data protection partitions the system into two domains—the secure domain and non-secure domain—using TrustZone technology incorporated in some ARM processors[*8]. The secure domain allows access to all terminal resources while the non-secure domain allows access to only specified resources. This access control can be forcibly applied at the hardware level. The OSTI specification assigns the secure domain to the Operator Domain and the non-secure domain to the Enterprise Domain. Adopting a configuration whereby the Enterprise Domain cannot access important data in the Operator Domain prevents crucial data from being abused even if the Enterprise Domain should gain OS kernel privilege. Although TrustZone technology is limited to ARM processors, similar results can be obtained by equipping other types of processors with a similar domain-separation capability.

4. VMM

This chapter describes the VMM as another method for implementing the OSTI specification. The VMM provides a set of virtual devices corresponding to hardware such as the CPU, memory, and peripherals and provides the execution environment for those devices to multiple OSes. The VMM in OSTI can execute multiple OSes in parallel on a single terminal and does not impose the OS Switching restriction that both OSes cannot run simultaneously. There are costs, however, associated with this virtualization processing—it increases the use of CPU, memory, and other system resources and increases power consumption.

The OSTI specification prescribes the following three items as an interface between the Enterprise Domain and DAL when using VMM.

- Enterprise OS launch, pause, and resume
- Enterprise OS system management interface:
  platform management, data communication between Operator OS and Enterprise OS, storage, peripheral devices
- VMM specific consideration:
  peripheral-device switching mechanism

An important feature of the VMM is

[*4] **VoIP**: Technology for sending and receiving audio data on an IP network.

[*5] **OS kernel**: Software that forms the nucleus of an operating system. It performs basic OS functions such as management of disk and memory storage and other computer resources, interrupt processing, and inter-process communications.

[*6] **TrustZone®**: A registered trademark of ARM Limited, UK.

[*7] **Reverse engineering**: A process of analyzing the configuration and operation of software or hardware to clarity manufacturing methods and operating principles.

[*8] **ARM processor**: A CPU that adopts architecture developed by ARM Limited, UK. ARM processors are widely used in mobile terminals and portable game devices.

its ability to control the switching and sharing of peripheral devices between OSes to maintain consistency as a mobile terminal even in an environment that runs two OSes in parallel. The OSTI specification divides peripheral devices into three classes: core (e.g., display, keypad), on-demand (e.g., camera, microphone), and shared (e.g., speaker, vibration device). Peripheral devices belonging to the core class are linked to the domain-switching button and switched all together to the foreground domain. In contrast, allocation of those peripherals belonging to the on-demand class can be switched to either domain in response to requests issued by applications in either of those domains. In this regard, a lock/unlock interface is provided so that domain switching can be performed and business applications executed while a call is in progress while at the same time switching the microphone to prevent voice input from being cut off. Finally, peripheral devices belonging to the shared class can be used simultaneously by both the foreground and background domains. Here, however, we can consider a situation in which silent mode has been set by either of the two domains. In this case, the other domain would likewise prevent the speaker from making any audible sounds.

To decrease resource consumption, which is an issue when using the VMM method, the OSTI specification also calls for an interface for placing the background domain into a suspended state and an interface for managing the power used by peripheral devices.

5. Conclusion

This article described the basic features of multi-domain architecture as specified by the OSTI specification and introduced OS Switching and VMM as two methods for implementing OSTI.

The OSTI specification is a significant development—it is the first in the world to specify a multi-domain architecture for mobile terminals that enables the use of existing application software developed by enterprises and individuals while maintaining the quality and security of traditional operator services.

For the future, we plan to develop prototype systems based on this specification for evaluation and testing purposes. We also plan to obtain technical feedback on the OSTI specification from a wide range of terminal manufacturers and software development companies with the aim of improving OSTI technology.

References

[1] Intel Corporation and NTT DoCoMo, Inc: “Open and Secure Terminal Initiative (OSTI) Architecture Specification Revision 1.00,” 2006.
http://www.nttdocomo.co.jp/corporate/technology/osti/index.html