A Multi-card Architecture for Smart Card Management Systems

Ryutaro Toji[†] *and Yoshinori Wada*

Abstract

Because there are many standards for smart card platforms, multiple types of smart card management systems must be introduced if a card issuer wants to issue and manage smart cards belonging to several platforms. Our network-based IC card environment "NICE" provides multi-card support that enables smart cards of various card manager (CM) types to be accommodated by a smart card management system by adding CM-dependent modules. It does this by localizing functions that depend on CM differences between platforms and concealing them from the upper-level system.

1. Introduction

As smart cards become more powerful with enhanced functionality and greater capacity, a multi-application smart card that can accommodate multiple applications and be used for multiple services is attracting considerable attention. The last half of the 1990s saw the coming of "smart card platforms" that provide operation and management functions for adding and deleting applications safely to and from multi-application smart cards. Examples of such platforms are MULTOS (Multi-application Operating System) [1] and Open Platform [2].

NTT has also been active in this area, developing in 2001 a network-based IC card environment (NICE) [3]-[5], which was chosen as the common smart card management system for multi-application smart cards in the "Research Project on Cities Equipped with Information Technologies" (hereafter abbreviated to CEIT), a field experiment held by Japan's Ministry of Economy, Trade and Industry from January to March 2002 in 21 regions throughout Japan involving one million contactless multi-application smart cards.

A smart card platform is significant because it can lower the development cost of a smart card management system by establishing specifications for the common functions needed for operating and managing smart cards as a system independent of hardware and applications. Nevertheless, it is not clear at this time what functions are really needed for smart card operations and management. For example, the range of specified functions and implementation methods varies greatly between MULTOS and Open Platform. A smart card may require highly advanced operations and management functions or relatively simple ones depending on the intended cost of the card and its business application area. When issuing and operating multiple types of cards, operators and system vendors will likely face a heavy burden if multiple types of smart card management systems must be introduced.

In view of these problems, there is a clear need for a mechanism that can accommodate cards of different platforms by localizing and exchanging those functions of the smart card management system that depend upon platform differences. NICE incorporated this idea from the start in the development of a "multi-card architecture" [6] mechanism. CEIT demonstrated the feasibility of this approach: two kinds of smart cards, each incorporating a different type of card manager (CM) could be accommodated in NICE.

This paper describes the NICE multi-card architecture.

† NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan
E-mail: toji.ryutaro@lab.ntt.co.jp

2. Overview of NICE

2.1 Concept of NICE

Figure 1 illustrates the concept of NICE, which is a general-purpose multi-application smart card platform that aims to support a service-oriented business model and network-based operations and management. The conventional business model was centered about the card issuer and assumed that the card issuer would usually select and load the card applications on the smart card when issuing it. In future, however, as the use of large-capacity multi-application smart cards spreads, users will be able to select card applications as needed from many service providers that have concluded contracts with the card issuer. Moreover, they will be able to download these applications onto their smart cards over the Internet from home or office. In short, the smart card business model of the future will be centered on services. NICE supports this service-oriented business model by offering a memory rental model, flexible business-role model, asymmetric-cryptography-based security framework, and remote operation over the Internet.

2.2 Architecture of NICE

Figure 2 shows a block diagram of the NICE archi-

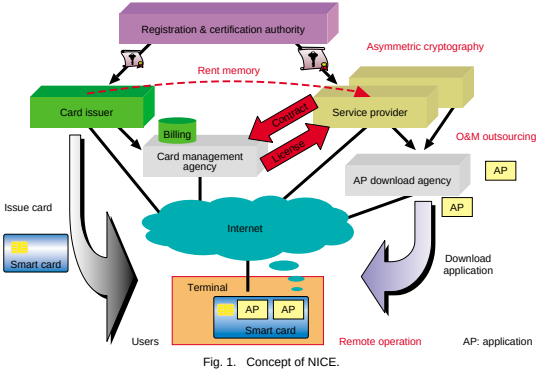


Fig. 1. Concept of NICE.

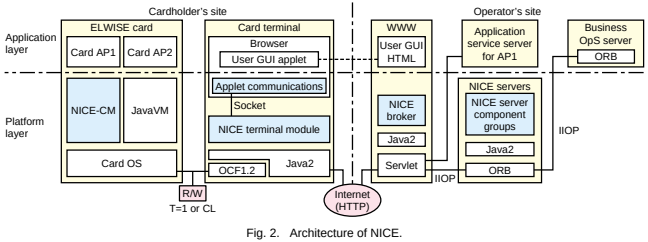


Fig. 2. Architecture of NICE.

tecture. NICE consists of a NICE-CM, a NICE terminal module, a NICE broker, and NICE server component groups (NICE servers).

NICE-CM is a privileged program on a smart card that manages the card and applications on it. The NICE terminal module resides on a terminal and is a basic module for controlling a reader/writer device and terminal/server communications. The NICE broker manages communications between the NICE terminal module and NICE server. It also provides special functions for supporting remote smart card operations such as managing a hot list of illegal cards and conducting operations based on that list.

The NICE server consists of many server components implemented as CORBA (Common Object Request Broker Architecture) [7] objects. They are classified into two groups: a management server group and an operation server group. The former provides individual management functions such as card status management, card memory area management, and license management and the latter executes actual operation sequences by calling individual management servers in accordance with operation requests submitted from a terminal. The management server group can also be called directly from a business operations system (OpS) when an operator sets or refers to some management data.

3. Issues in multi-card support

Multi-card support requires:

① A method that enables the smart card management system to identify different CM types.
② A common interface for concealing differences among CM types.

Below, we examine these issues by taking as an example two CM types, namely NICE-CM and X-card-CM that were accommodated by NICE in CEIT.

3.1 Identifying the CM type

When a smart card is inserted into a reader/writer connected to a terminal, the system must be able to determine the CM type of that card so that it can initiate the corresponding processing, but there are no standard specifications for doing this. Historical Bytes of ATR [8] or ATTRIB [9] are not suitable for this purpose, because their usage is not standardized and could be rewritten by applications. In CEIT, this problem was solved by storing the CM-type ID in the CM, establishing a uniform application ID (AID [10]) in the CM, and having each type of CM support the same GET DATA command that can extract the CM-type ID from a CM. In practice, however, it is generally difficult to unify among platforms the AID in CM and specifications for CM commands/responses for obtaining CM type. A more realistic method might be to standardize a common card application for storing the CM-type ID and processing a common GET DATA command and then loading the application on cards of each platform through an identical AID. A smart card management system can distinguish the CM-type of a card by selecting this common application and executing the GET DATA command.

3.2 Common interface

For a smart card management system to accommodate cards of different CM types, the system must be divided into a common section independent of the CM type and individual sections dependent on CM type, and a common interface independent of CM type linking these sections. While the CM external interface is usually specified in APDU (application protocol data unit [8]) commands and responses, these generally differ between platforms. An individual CM section must generate a command APDU specific to that CM type based on data received from the common section via the common interface, and it must also analyze the response APDU and return results to the common section. The problem here, however, is what level of abstract processing units is appropriate for the common interface. Specifically, two levels can be considered: an operation level visible to the end user for performing functions like downloading and deleting applications, and a CM function level like mutual authentication and PIN (personal identification number) checking, which are performed in some operation processes.

Figure 3 compares the command sequence for downloading applications between the two types of CMs accommodated by NICE. Based on this comparison of command sequences and command formats between two types of CMs, we concluded that creating a common interface at the function level is very difficult. The common interface should be designed at the operation level, while the command format, execution sequence, and generation of card-type-dependent data should all be carried out in card-type-dependent sections.

4. Implementation of NICE multi-card support system

4.1 Architecture for multi-card support

Figure 4 outlines the NICE multi-card support

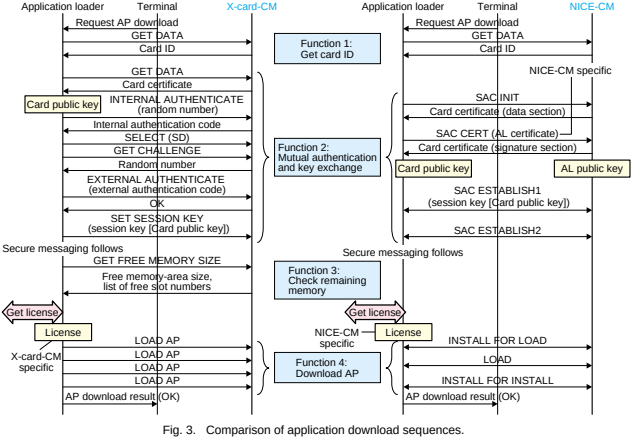


Fig. 3. Comparison of application download sequences.

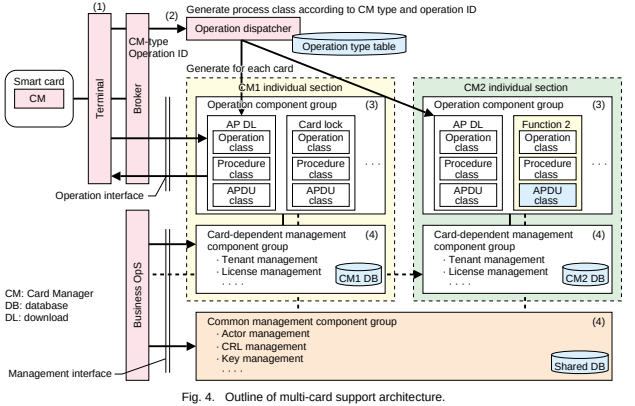


Fig. 4. Outline of multi-card support architecture.

architecture. It consists of four main elements.

(1) Terminal-based functions for identifying the type of card inserted into a reader/writer and the type of operation specified by the user and conveying them to the server. Table 1 lists operations defined in NICE.

(2) A server-based operation dispatcher that receives the card and operation from a terminal and selects and initiates an appropriate operation server component to execute the operations requested by the user.

(3) An operation component group that exists for each card or operation. This group provides commands and data according to card-dependent control procedures and executes requested operations by sending and receiving commands and responses to and from the smart card.

(4) A management component group that generates and stores data required for operating and managing cards and card applications. This group consists of a card-dependent management component group that manages data dependent on card type and a common management component group that manages data independent of card type.

Operation components come in three class layers: operation layer, procedure layer, and APDU layer. The operation layer manages card operations visible to the end user such as downloading applications. On this layer, the operation dispatcher sends requests to class factories for each CM type and generates classes corresponding to each operation. This layer implements common processing independent of CM type. A new operation can be added by simply adding a new class to this layer. The procedure layer manages procedures for card control. It implements processes that depend on CM type such as APDU sequences for downloading applications. The APDU layer generates command APDUs and analyzes response APDUs. A new APDU command can be added by simply adding a new class to this layer.

There are two kinds of common interfaces for multi-card support. One is an operation interface between the terminal and operation server components for sending and receiving information neces-

Table 1. NICE operations.

Operations	Description	Common
AP download	Download card application	○
AP delete	Delete card application	○
AP tenant change	Change memory size available to card applications	
SD generation	Generate service domain	
SD change	Change size of service domain	
SD delete	Delete service domain	
Card log reference	Reference card log	
AP profile reference	Reference card-application attribute information	
SD profile reference	Reference service-domain attribute information (AP list etc.)	○
Card status set	Set card lifecycle status	○
Card profile reference	Reference card attribute information	○
Card profile change	Change card attribute information	
CI certificate update	Update card-issuer public-key certificate	
SO certificate update	Update service-domain-operator public-key certificate	
Card certificate update	Update card public-key certificate	
SO public key update	Update service-domain-operator public key	
SP public key update	Update service-provider public key	
AP lock	Lock card application	○
AP unlock	Unlock card application	○
Card status get	Get card lifecycle status	
PIN cancel	Cancel user personal identification number	○
PIN authentication	Authenticate user PIN	○
Service execution	Execute card application using NICE terminal	

○: Common operations supported by both NICE-CM and X-card-CM.

sary for an operation and APDU commands and responses to and from the card. The other is a management interface between the business OpS and management server components. This interface enables the business OpS to set and read necessary data concerning of management components.

4.2 Accommodating other CMs

Many operations are defined for NICE-CM (Table 1). X-card-CM supports some of them and some of its own unique operations that are not included in the table. These nine basic operations indispensable for multi-application smart card management in the project were selected and implemented, as indicated as common in Table 1.

NICE that accommodated X-card-CM was deployed in seven regions of Japan in CEIT and proved to be a useful system for performing various smart-card tasks including issuing cards and downloading and deleting applications.

5. Conclusion

NTT's NICE multi-card architecture enables smart cards of various CM types to be accommodated by a smart card management system by adding CM-dependent modules. This is done by localizing functions dependent on CM differences between platforms and concealing these functions from the upper-level system. In the Research Project on Cities Equipped with Information Technologies held in Japan, smart cards incorporating two CM types were successfully accommodated in NICE. Moreover, this architecture was chosen as the standard smart card management system for the national "Basic Residential Register Cards" [11], which are scheduled to be issued to residents by local governments from August 2003. It lets local governments select optimal residential cards that have different types of CMs and accommodate them in a smart card management system.

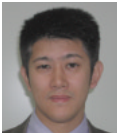
There are two main issues that need to be addressed other than the CM identification method. The first concerns the need to share management components and establish a common management interface for the business OpS. In the architecture proposed here, all management components related to CM-dependent data result in a design that must be maintained for each CM. If good commonality can be achieved in the management interface between CM types, it might be possible to achieve some of the management functions now achieved by CM-dependent components in common components.

The second issue concerns the need to support multiple encryption systems. This is not a problem of NICE multi-card architecture itself—it is a matter of practicality.

At NTT, we plan to expand multi-card support by accommodating CMs of other platforms like Open Platform or MULTOS in the future.

References

[1] MULTOS, http://www.multos.com/
[2] Open Platform, http://www.globalplatform.org/
[3] R. Toji, Y. Wada, S. Hirata, and K. Suzuki, "A Network-based Platform for Multi-Application Smart Cards," Proceedings of fifth IEEE International Enterprise Distributed Object Conference (EDOC2001), pp. 34-45, IEEE Computer Society Press, 2001.
[4] R. Toji, Y. Wada, S. Hirata, and K. Suzuki, "NICE—A Network-based Platform for Multi-application Smart Cards," NTT REVIEW, Vol. 14, No. 1, pp. 13-19, 2002.
[5] R. Toji, Y. Wada, S. Hirata, and K. Suzuki, "NICE: A Service-oriented Platform for Multi-Application Smart Cards," Proceedings of e-Smart 2002, pp. 45-56, Sep. 2002.
[6] Y. Wada, H. Akashika, and R. Toji, "A proposal for accommodating different card managers in one smart card platform," TECHNICAL REPORT OF IEICE, KBSE 2001-46, IEICE, pp. 9-16, Dec. 2001.
[7] CORBA, http://www.omg.org/
[8] Information Technology, Identification Cards, Integrated Circuit(s) Cards with Contacts, Part 4: Inter-industry Commands for Interchange, ISO/IEC 7816-4, 1995.
[9] Information Technology, Identification Cards, Contactless Integrated Circuit(s) Cards, Proximity Cards, Part 3: Initialization and Anticollision, First Edition, ISO/IEC 14443-3, 2001.
[10] Information Technology, Identification Cards, Integrated Circuit(s) Cards with Contacts, Part 5: Numbering Systems and Registration Procedure for Application Identifiers, ISO/IEC 7816-5, 1994.
[11] http://www.ntt-neo.com/english/news/030401.html



Ryutaro Toji
Senior Research Engineer, Supervisor, Producer, Planning Section, NTT Information Sharing Platform Laboratories.
He received the B.S. and M.S. degrees in basic science from the University of Tokyo, Tokyo in 1985 and 1987, respectively. In 1987, he joined NTT Information Processing Laboratories, Tokyo, Japan. Since 2000, he has been engaged in research and development of smart card management systems. He is a member of the Institute of Electronics, Information and Communication Engineers.

Yoshinori Wada
Smartcard Service Platform Project, NTT Service Integration Laboratories.
He received the B.E. and M.E. degrees in electric power engineering from Musashi Institute of Technology, Tokyo in 1995 and 1997, respectively. In 1997, he joined the Software Laboratories, NTT, Tokyo, Japan. He is a member of the Information Processing Society of Japan.