PAPER

Smart Card Information Sharing Platform towards Global Nomadic World

Eikazu NIWANO[†a)], ***Member***, **Junko HASHIMOTO**[†], ***Nonmember***, **Shoichi SENDA**[†], **Shuichiro YAMAMOTO**[††], ***and*** **Masayuki HATANAKA**[†], ***Members***

SUMMARY The demand for multi-application smart card platform has been increasing in various business sectors recently. When it comes to the actual implementation of the platform, however, network-based dynamic downloading in a Card Issuer-Service Provider separated environment has not made much progress. This paper introduces the smart card information sharing platform that uses licensing/policy/profile management and PKI-based technologies to enable multiple CIs and multiple SPs to reflect their own business policy flexibly via network. It makes the paradigm shift from card-oriented scheme to service-oriented scheme. By through world's first implementation of the scheme and some experiments including deployment, we confirmed that this technology is well-accepted and applicable to various business sectors and it can be of practical use.

key words: *smart card, platform, multi-application, PKI*

1. Introduction

To support smart card operation management, the demand for multi-application smart card platform technology has been increasing in various business sectors recently. Especially Card Issuer(CI)-delegated Service Provider(SP) dynamic download service in a distributed environment has become a focus of attention, rather than the type of CI-driven application(AP) local dynamic download onto a card. The new framework, however, is still static in terms of types of delegation of authority and its workflow that is done from CI to SP. This has been an issue that prevented dynamic operation management in accordance with business schemes. For example, there were some inconveniences for SP. SP could install only the limited number and types of services that specific CI has selected, so SP could not start their original smart card services easily at lower cost. And SP could not cooperate with multi-CIs.

To resolve it, we have studied this issues and worked with Gemplus [1], [8] on smart card information sharing platform called NiNa (Nomadic Information-sharing Network Architecture) by using license/policy/profile management and PKI-based security technologies that enable flexible and trusted smart card management among multiple CIs, SPs and Card Holders(CHs). We also developed the prototype with 2 types of card, one is ELWISE [12] card by NTT which has 1 MB flash memory, and 13 applications such as electronic money, electronic passport, and electronic loyalty etc. We exhibited it at world's largest smart card exhibition Cartes2000 in October 2000, as the world's first implementation and a pioneer system which realized CI-SP separated authority delegation model based on PKI technology via network.

On the other hand, NICSS (the Next generation Ic Card System group) [2], who is working on the standardization mainly for public sectors in Japan, standardized the first draft specification including the model in April 2001 based on our proposal. And then the Ministry of Economy, Trade and Industry (METI) of Japan started "The Research Project on Cities Equipped with Information Technology (IT CITY) [3]" based on the NICSS framework from January to March, 2002. Based on NiNa concept, NTT individually developed the commercial use product called NICE (Network-based Ic Card Environment) [10], [14] and the product was deployed in the IT CITY project. And also we proposed NICSS framework to eEurope/Smart Card Charter/TB7(multi-application)/WG4(multi-application architecture) as a PKI-based cross-sectional generic multi-application platform for multiple card communities and it was adopted [32]. The eEurope Smart Card (eESC) initiative was launched by the European Commission in December 1999 as an immediate outcome of the eEurope initiative [33].

In this paper, we would like to describe the NiNa conceptual/design model, prototyping overview, and evaluation based on questionnaire, comment through demonstrations, and above mentioned deployment. Section 2 describes existing models on smart card platform. Section 3 mentions the NiNa technology to overcome existing problems. Section 4 describes some concrete effects of this technology. Section 5 describes an implementation developed as prototype. Section 6 discusses the evaluation of utility, practicability and applicability. Section 7 refers the related technologies.

2. Nomadic World

From a point of view of standardization and models, Master Card-driven MULTOS [4] and Visa-driven VOP(Visa Open Platform)/OP(Open Platform) [5] managed by GP(Global Platform) [6] are the two major specifications.

MULTOS adopts token-based operation right management for its standard. This platform, however, takes no considerations about authenticating SP and so can be said CI-

Manuscript received June 30, 2003.

Manuscript revised October 8, 2003.

[†] The authors are with NTT Corporation, Musashino-shi, 180–8585 Japan.

[††] The author is with NTT Data Corporation, Tokyo, 104–0033 Japan.

a) E-mail: niwano@ntt.fr

centric framework. The reason is that MAOSCO Ltd. basically controls all the card management in accordance with the company policy.

GP is promoting standardization of dynamic download which supports online AP download and token-based operation right management, but it adopts symmetric key-based framework by card issuer for mutual authentication between card issuer/service provider server and card. This implies the service provider can only download their applications to specific cards whose issuer is the same. In other words, card holder who belongs to some card issuer cannot download and cannot use applications whose service provider belongs to other card issuers. Thus each card community which is composed of a card issuer, card holders and service providers are isolated and the download sequence or work flow is still card issuer-oriented one based on specific business process such as card issuer-centric financial area typically.

But in near future, the number of service providers will increase and we believe they want to distribute their applications dynamically and securely to many cards whose card issuers are different. As well as service provider, we are sure card holder would like to use any service anytime anywhere beyond card community. For instance as a typical case in e-governmental sector a citizen who has citizen card of some city must be able to download city information application of other city to their citizen card and use city information service when he/she visits there dynamically. Furthermore these business model and process should be defined and executed according to the business policy of each other flexibly and dynamically.

To realize such free and trusting global nomadic world by using smart card and smart card platform is our objectives.

3. Smart Card Information Sharing Platform

3.1 Concept

The goal of Smart Card Information Sharing Platform NiNa is to provide architecture applicable to a wide range of business schemes - models and processes - autonomously. It means stakeholders who participate in the smart card business can cooperate with each other flexibly, dynamically and trustingly based on their business policy or demand.

Especially, service provider does not need to issue and manage the cards. And they should be able to concentrate themselves on provisioning with their services. Also they can be independent of one specific card issuer and cooperate with other "multiple" CIs/SPs as they like, in order to provide their services flexibly and widely. This means this architecture supports various business models and processes ranging from CI-driven models, in which CI download applications onto a card of CH, to SP-driven models, in which CH access to SP to download application services on their demand. Smart card platform is going to be changed from CI-driven card-oriented system to SP-driven "service-oriented" one, which we expect a large number of SPs to join in and to diversify the services.

And also based on the idea that a smart card is designed as an agent of a card user for managing his right to access to all the resources in both real and virtual worlds [16], [17], it is clear that flexible definition and management of authority among the entities in this business is very essential. As the ultimate goal, CH needs to become an entity independent of CI and SP and to be supported as a card "owner". In that case, card owners can also select and download any applications into their own card securely on the basis of CH-driven dynamic contract with CI and SP. For example CH who has not registered to any CI can download application from unregistered SP by dynamic negotiation like impulse buying. This may be called as "strongly dynamic download" in comparison with conventional dynamic download. Then, just we can say it also makes paradigm shift to "user-oriented" smart card system.

Finally because NiNa supports establishing a secure network that enables these stakeholders interoperable through network anytime, anywhere, NiNa can make paradigm shift to "network-oriented" smart card system, moreover. So let us summarize the requirements [9], [29] for NiNa.

- Separation of SP from CI: Needs to enable card users and SP to use and provide services of their own will, which means SP must be independent of CI and manage her card operation. This reduction of SP's dependence on CI gives SP flexibility.
- Flexible card control after card issuance: Needs to allow SP or CI to remotely update information on card access and execution control and also lock a card when it is lost or unauthorized access comes in. Put simply, operation policy can be changed dynamically even after a card is issued. This profile-based policy control enables CI to flexibly control card operation, namely CI permits and prohibits any specific application downloads, maintains expiration date and valid frequency, and gives CH a warning when memory resources are running out.
- Cost sharing: Needs to have card hosting service like server hosting in computer world. That means SP can rent CI's memory domain in accordance with business model. When an application is downloaded into smart card, for example, SP pays CI the rental fees for using tenant. With this scheme, CI can start a new business of card hosting by renting card memory space to SP. This will set SP free from all the card related operations like the issuance and maintenance, and it also allows SP to provide their services at lower costs.
- Secure transaction: Needs to execute secure transaction between card user and SP, to be more specific, between card and server. No unauthorized access, tampering and data leakage are allowed. This makes CH no worry about downloading unauthorized or corrupted applications. Needs to get back to a normal operation when communications and any other types of errors occur. Needs to keep secure transaction in case a dispute arises between CI and SP.

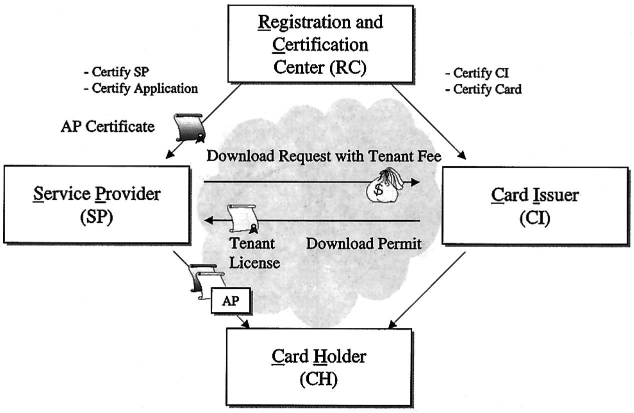

Fig. 1 Role model with an example of typical download sequence.

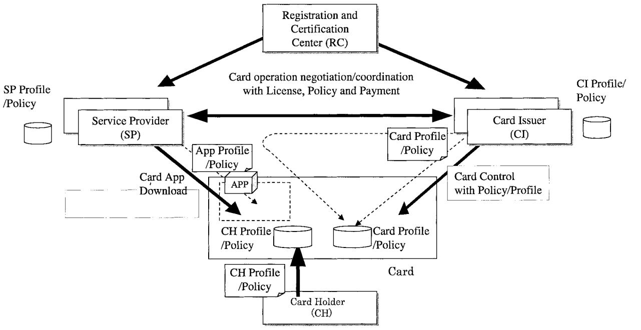

Fig. 2 Dynamic and flexible control using policy and license with profile.

- Optimum Access: Need to allow card users to access to SP or CI without noticing the ways and procedures they are using. That means users do not need to notice anything about the types of smart cards, reader/writer, terminals and network.

3.2 CI-SP Separated Basic Role Model

To satisfy the 1st requirement, NiNa placed Registration and Certification Center (RC) who knows and authorize not only applications/cards but also CI, SP by PKI-schema (See 3.4 in detail). It enables NiNa provides with a distributed system architecture to enable multiple CIs/SPs to cooperate with each other separately. This means that NiNa enables each CI/SP to manage other multiple SPs/CIs respectively based on their contract.

As Fig. 1 shows the primary roles of NiNa includes 4 roles such as RC, CI, SP and CH. Secondary roles such as downloader who are outsourced to manage download, platform provider who are outsourced to manage all the functions from those 4 roles, application provider, card manufacture can be considered but we only refer primary roles in this paper.

Let one NiNa system be composed of one RC and several CIs/SPs/CHs, and NiNa network be composed of several NiNa systems under those federated RCs. A card community composed of one CI, multiple-SPs and CHs can be inter-operable in the same NiNa system and NiNa network.

3.3 Network-Based Policy Management and License Control

To satisfy the 2nd to 3rd requirements, we introduced a mechanism of network-base licensing and policy management using profile as Fig. 2 shows. Licensing is a mechanism to delegate operation right to a third party. Policy control is a mechanism to manage control rules for the events arising from resources maintained by each role player (actor; stakeholder). The policy information is used to manage licensing (issue a license) and execute the licensing related transactions. By maintaining these licensing and policy management via network, the rules of access and execution control required for card operation in card can be changed dynamically.

This mechanism allows CI and SP to independently establish their policy and separate/delegate operation right so that card operation in line with business scheme becomes feasible. Billing policy in each contract defines the details of billing. And also delegated card operation or card policy management can be done through SP node using this framework (See Fig. 2). This means lock of card in case of irregular usage of it can be realized.

3.4 PKI-Based Security Framework

To satisfy the 1st to 4th requirements, we use asymmetric cryptography for mutual authentication between CI/SP and card. We also use symmetric cryptography for encryption via network. In this model, RC issue public key certificate to those CIs/SPs and card/application certificate. It enables us complete secure framework among multiple CIs/SPs through RC. To recover the communication error in case of card operation, NiNa supports distributed transaction management function. To cope with the disputes, it uses audit trail management. With these functions, network-based secure card operation can be executed.

3.5 Adapter-Based Communication Control

To satisfy the 5th requirement, NiNa provide with an adapter for the protocol handlings and adaptive control. With this adapter, various types of media such as low-end to high-end of cards, personal computers, mobile phones, Internet and telephone/mobile network and various types of protocol between card and server or security protocol including security protocol can be handled. And also in cooperation with above mentioned techniques, card can cooperate with CI server through card-SP server line and SP-CI line even if direct connection to CI does not exists.

4. Example of Major Effectiveness

By using this model, following flexible card management can be realized.

- Strongly dynamic download: SP can download their application to someone's card on CH's demand, even if SP has not known who CH are and who the CI is, as we mentioned earlier. Because SP can get the identity information or profile information of CI/CH from card. And also SP get the detailed profile of CI including address of CI and public key from RC dynamically, and communicate with CI and can make decision after the coordination of CI/SP policies. Thus the combination of policy & profile management function and PKI-based framework enables SP/CI (even if it is CH) to authenticate and coordinate each other dynamically, even if the contract among them has not been made before.

- White card model support: In the above case, if the card policy does not have any constraints, it means CI does not require anything from SP, SP can download applications to the card even if they do not have the license from CI. Nevertheless, trust of the application is guaranteed by RC.

- Online/Offline control: Such licensing mechanism enables service providers/card holders to download applications in offline environments, because the SP can get licenses or certificate from CI and RC in advance. Or policy control mechanism enables the same thing because card holder/card issuer get/put card management policy regarding permission to some service providers and some types of applications in the card in advance.

- One time or period-restricted application: Application Control policy can describes the numbers or period of execution of application. Then for example 1 time application and period-restricted application can be realized.

5. Prototyping

We completed the NiNa prototyping in October, 2000 in collaboration with Gemplus. It is the world's first implementation and a pioneer system that realized secure online card application downloading with PKI-based mutual authentication framework in a CI-SP separated environment. This supports multiple CIs and SPs collaboration. We used Java technology only.

5.1 Functions

As shown in Fig. 3, six functions are supported for SP operator menu. Those are Card Issuer Management, Card Holder Management, Application Management, Policy Management, Billing Management and Audit Trail Management. The card issuer management is used to record multiple CIs who provide rental card spaces. The card holder management is used to administrate CHs who use APs that are provided by SPs. The application management is used to manage downloadable applications. The billing management is provided to see billing information at time of downloading. The audit trail management is provided to keep logs of transactions. The Policy Management menu is provided to change operation policy such as card operation. The same kinds of menus are provided to CI. And also we developed some menu including viewer of multiple applications in a card for CH.

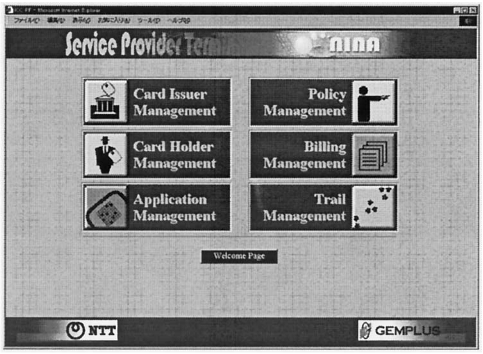

Fig. 3 Operator menu of service provider.

5.2 High Performance JavaCard

By extending JavaVM reference [7], we have developed "Sapphire" [11], [13] JavaVM that runs on an ELWISE card that was also developed by NTT as the world's first smart card having 1 MB of nonvolatile flash memory.

The JavaCards now available in the market have the upper limit of 32 Kbyte memory capacity and downloadable applications are limited in number even though the Cards are called multi-application smart cards. When we want to implement Card Manager that provides a wide range of functions such as multi-application management or AP online download, the cards requires 20 Kbyte memory space at least. If we download the Card Manager onto the currently available JavaCards, the number of APs and the size of the Applications become small. Therefore we need to have a smart card having larger memory space like Sapphire. The Sapphire is designed to resolve these issues, providing a large 1 Mbyte of flash memory capacity implemented on an ELWISE card.

The ELWISE is a contact-type multi-purpose smart card. Equipped with special circuitry capable of handling multiple encryption schemes and a large nonvolatile memory (more than ten times the size of conventional card memories), the card can be used for a wide range of services such as electronic payments, multimedia data communications, and medical applications. The card allows a number of conventional single-purpose card applications to be integrated into a single ELWISE card.

5.3 System Configuration

NiNa supported 2 types of JavaCards. One was above-mentioned NTT-developed Sapphire with 1 MB of flash memory implemented on an ELWISE [12] card. The other was Gemplus-developed GemXpresso [1] card having 32 KB of EEPROM. 10Base-T LAN was used to connect terminals and servers. Dialup connection using PHS and LAN to LAN connection using ISDN has been provided.

Card manager (CM) was implemented on the cards to communicate with the servers for card management. The

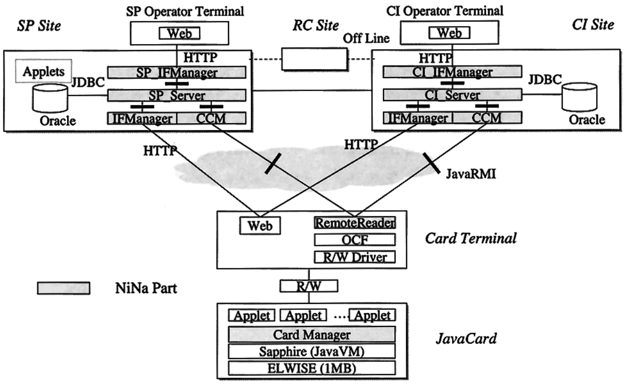

Fig. 4 Prototype system configuration.

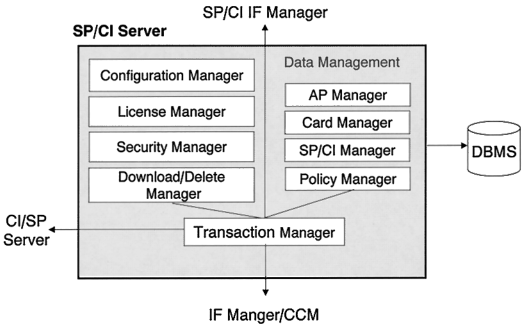

Fig. 5 Software components in SP/CI server.

functions include CM status management, AP management (including download function), policy management, security management, transaction management, PIN (Personal Identification Number) management, and shared information (profile) management.

The terminal-installed remote reader transmits and receives APDU (Application Protocol Data Unit) [30] messages between the servers and OCF (Open Card Framework) [31].

As Fig. 4 shows servers are constructed with 3 layers; 1) IFManager/CCM that functions as a gateway between user terminal and server, 2) CI/SP servers that manage cards and applications, and 3) CI/SP_IFServer that serves a gateway between operator and server. CCM (CardCommunicationManager) is an "adapter" that refers to card profile information to identify which type of JavaCard (as a simple case of 3. 5) is being used, and controls communications depending on the card type.

HTTP is used for Web terminal interface including CH, SP and CI. JavaRMI was used for the communications between the CI and SP servers and between the servers and terminals. JDBC was employed for database communications. Communication between CI/SP and RC is done off line at the prototype.

SP and CI use almost the same structure. The difference is that SP processes license request while CI issues the license. As Fig. 5 shows, CI has a function of card management that maintains card type information and issued card information besides card issuance tools. SP has a function of application management that maintains the types of SP card applications. Multiple contractors can be maintained in each of the server. To be more specific, SP can maintain multiple CI and vice versa. With regard to download license, SP sends the download request to CI every time SP needs it. In respect to renting a card space (tenant management), we created the space for only applications required at time of download. Our product-version implements the actual memory domain.

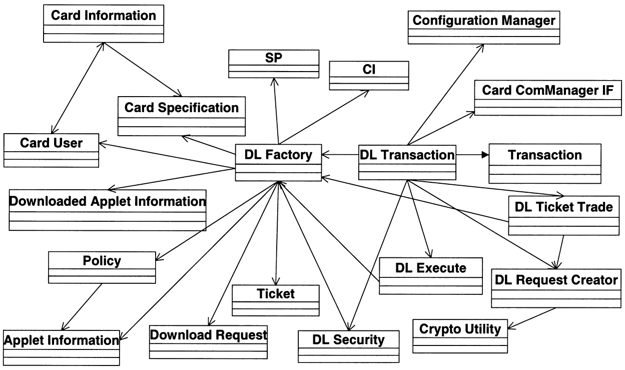

Fig. 6 Class overview of SP server.

We developed 13 applications such as e-passport, e-air ticket, e-loyalties, e-money, e-medical record, e-traffic card etc. And all of them were stored on one ELWISE card. Total size of applications is about 100 KB [13].

5.4 Class Components

Figure 6 shows the class diagram overview of the functional module called "SP server" in Fig. 4. Those classes correspond to the subdivided functional modules of "SP server" described in Fig. 5. As mentioned in 5. 1, CI has almost the same server structure as SP's. A component of SP's DLRequestCreater (request download license) corresponds to the CI's DLTicketCreate (issue download license). For the security, we used RSA/T-DES stored in the NTT-developed ISEC library. The term 'ticket' in Fig. 6 means 'license' in Fig. 5.

5.5 Download Sequence

Figure 7 shows an overview of the download sequence. First, when CH sends a download request to SP, SP server and card start using an asymmetric cryptography (RSA) to authenticate the CH. After the authentication, the session key is generated and exchanged within the same session and the key is going to be used for all the encryption within the same session. To speed up this set of security transactions, SAC (Secure Authentication Channel) was developed and implemented for the NiNa prototype. The Characteristics of SAC are as follows; 1) adopt challenge and response type mutual authentication between CI and SP using public key certificate, 2) define reduced format of X509 public key cer-

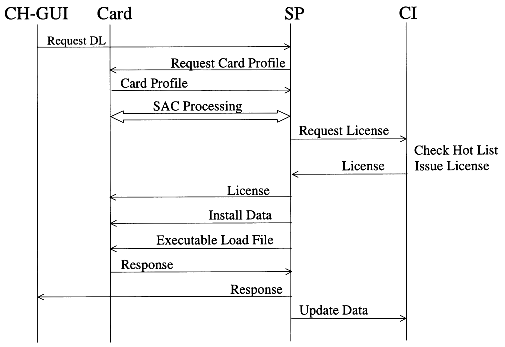

Fig. 7 Download sequence.

tificate by extracting minimum set of data in order to adapt it to the card such as Java Card that the size of communication data is restricted, 3) reduce the number of commands from 8 to 2 by combining mutual authentication with key exchange and sending multiple data at the same time in comparison with using ISO7816-4, ISO7816-8 [30].

Next, when SP requests CI to delegate a download authority, CI delegates it (licensing) to the SP (issues the download license). With this download license and RC-issued application registration certificate, SP starts downloading the card application onto a smart card. After confirming the download license is delegated from CI, the card starts downloading the application. The on-card transaction management copes with network failure and ensures secure download. SP uses a billing management function to pay CI for the rents of CI card memory domain (as a download fee).

5.6 Operation Management

SPs and CIs can remotely manage their applications and information on smart cards independently. CIs also can lock a smart card when it is lost or unauthorized access comes in. These kinds of erroneous smart cards are managed by a hot list. By using policy operation rules, it can be changed dynamically after a smart card issuance. Examples of policy rules are as follows.

Ex1) If valid(SP_Id) then issue_license(SP_Id).

Ex2) If invalid(Card_Id, Hot_list) then refuse(Card_Id).

These kinds of policy rules are simple but useful tool to describe card management operations.

CI manages all the information of issued smart cards and CHs. It also manages downloadable applications and the relationship between smart cards and applications.

Secure transaction between CHs and SPs is necessary to provide. No unauthorized access, tampering and data leakage are allowed. It is necessary to get back to a normal operation when communications and any other types of errors occur.

6. Evaluation

In this section, we will report on the functional and practical evaluations of the NiNa technology.

In the functional evaluation, we would like to estimate the utility of the NiNa concept or its functions by such as; 1) the multi-application environment, 2) the model in which SP is separated from CI, and 3) application download. And also we will touch on the utility for application developer.

In the practical evaluation, we would like to run a download performance test to see the usability.

6.1 Procedure

(1) Questionnaire by telephone interview
We conducted a telephone survey by questionnaires to a total of 52 organizations in the fields of communications, loyalty/retail, government/health, finance, universities, public transportation and others in European countries (the UK, France, Spain, Germany, etc). In this questionnaire, 1(very negative)-5(very positive) scale was used to evaluate the NiNa concept/functions. This investigation was done by March 2001.

(2) Feedback through demonstration
NTT and Gemplus jointly participated in "Cartes2000", the world's largest smart card exhibition held in Paris (France) from October 24th through 26th 2000. Besides this telephone interview, we have been collecting feedback data on the NiNa technology since then by exchanging views, participating in other exhibitions and conferences. It includes Smart Card 2001 (UK), Multi-application Card 2001 (Germany), JavaOne (USA) and Business Show 2001 (Japan).

(3) Performance evaluation
We used this prototype to measure the total time required for AP download from SP server onto card and the AP deletion from card respectively. The values were measured from when a card user presses the AP download button on the Web screen up to when the user sees the result on the screen. In this test, we checked two points: one is to see the time taken when the size of AP changes, and the other is to see how much time is allocated to each of the AP download process.

(4) Deployment
We tried to propose this model to adopt NICSS (the Next generation Ic Card System group) who is working on the standardization mainly for public sectors in Japan.

6.2 Results

(1) Questionnaire
Table 1 shows the score results on the NiNa technology by business domain. We used the score from 1 to 5; 1 is 'very negative' and 5 is 'very positive'. We calculated the average of these scores. The symbols in the table describe as follows; A: equal to or less than 5.00 and greater than or equal to 4.00, B: less than 4.00 and greater than or equal to 3.00, C: less than 3.00 and greater than or equal to 2.00, D: less than 2.00 and greater than or equal to 1.00?

We see from Table 1 that online AP download, especially downloading from various terminals on the way, is in high demand in loyalty/retail, public transportation, univer-

Table 1 The results of telephone interview regarding this model.

Items	Telecoms	Loyalty /Retail	Government /Health	Finance /Banking	Universities	Transport	Others
The number of samples	3	5	10	22	1	2	9
Total concepts	B	B	C	B	B	B	B
For application download	C	A	A	B	B	A	A
For data download	C	B	A	B	B	B	A
Download at home	C	A	A	C	D	A	A
Download on the way	B	B	A	C	A	A	A
SP gets authorization from CI	B	B	B	B	A	B	B
SP gets authorization from 3rd Party	C	B	C	C	A	C	C
Outsource card domain management	B	B	C	C	B	A	B
Outsource downloading management	B	A	C	B	B	A	C
E-contracts can change&policies	A	B	B	C	B	B	C
Recover cost of production	B	B	A	B	B	B	A
Billing and payment management	B	A	C	B	A	A	B
Audit trail management	B	A	B	B	A	B	B
It works with any network	A	A	B	B	B	A	B

sities and government.

In respect to delegating card management to a third party (authority delegation includes) like in the case of white card, mainly finance sector answered in the negative to the idea including the responsibility issue. As for card management outsourcing, the administrative and finance sectors responded negatively. This is mainly because they are so sensitive to secure information management that they want to keep the information under their control. Evaluation was divided with regard to policy management, and finance sector is especially very sensitive to the ways of dealing with applications. We could hear the request for the customized applications. The cost sharing concept was highly evaluated that uses license to sign a contract with CI as a business partner. We also received a positive response to network adaptability. However it was pointed out that the issues are not technical but business or political aspects like the development costs and keeping pace with infrastructure around them.

With regard to multi-application, most of the respondents said it difficult to combine medical and public transportation families and to combine the administrative (public) and private sectors. The respondents also shared the same view that the market for multi-application will be 4 to 6 years ahead of us from then.

(2) Comment feedback through demonstration

We collected almost the same comments as described above in and outside of exhibitions and other occasions, and we must pay attention to such skeptical opinions on general-purpose multi-application card.

NTT and Gemplus jointly participated in "Cartes2000", the world's largest smart card exhibition held in Paris from October 24th through 26th 2000, where we received an excellent response. "Le Monde", a French newspaper wrote an article about our prototype system entitled "Our dream of making the smart card our next generation computer" in their November 6th issue [18]. In the article, Mr. Michel Alberganti commented, "Our long awaited dream may not come true tomorrow, but it is just around the corner that piles of plastic cards and coins stuffed into our wallets will be totally replaced by ONE smart card".

(3) Performance evaluation

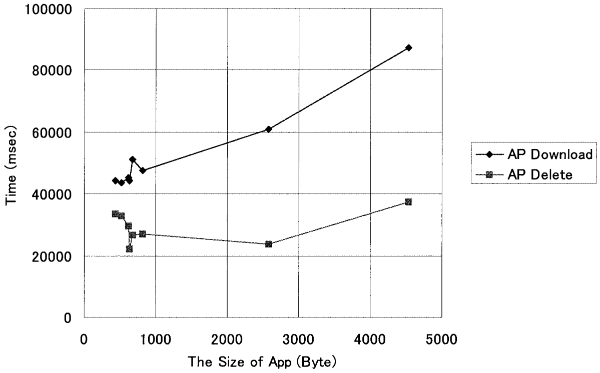

Fig. 8 The time required to download/delete card application.

We estimated the time of downloading/delete application according to the size of application. In this experiment, 8 different sizes of applications which have 0.43 kByte to 4.54 kByte respectively were prepared. Figure 8 illustrates the time for AP download process increases in proportion to the AP size, but the time required for AP deletion process makes no difference by AP size. This is because the frequencies of APDU transmission increase according to the size of AP that is downloaded. Under a 10Base-T LAN environment, it takes about 40-50 seconds (a little bit faster when log generation stops) to download a 1 kByte of application.

We also tried to clarify which module was bottleneck as Fig. 9 shows. To estimate it roughly, we calculated the ratio of the average of processing time of each module for total amount of those modules. Those modules are SP server, CI server, CCM, Remote Reader and Card + Alpha (Alpha: The modules between Remote Reader and Card as you can see in Fig. 4). This estimation is done for the above mentioned 8 different sizes of applications. To investigate it, we embed time counter in each module for data in and data out. But for card, we cannot see log in it, so we implied the processing time from Remote Reader log as the processing time. The total amount of the time was almost the same as downloading time, so we can take it as the ratio for downloading time in this experiment.

The figure indicates the transactions on the servers (SP server, CI server and CCM) account for 1/3 of the whole indicated processes. Load and install of an application onto

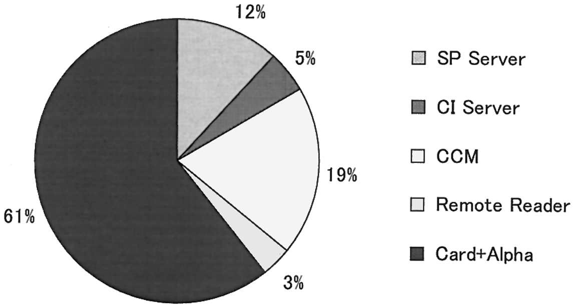

Fig. 9 The ratio of processing time for each module in downloading.

a card, on the other hand, makes up a little less than 2/3 of the whole process. We also found out that the processing time of SP server, CI server and CCM take almost constant time not according to the size of Application. The ratio of C + Alpha is between about 55% and 70%.

And this time allocation pattern can also be seen in the AP deletion process, and transactions on card side accounts for more than 1/2 of the whole processes and this tendency is almost the same in all the 8 applications, in other words, not according to the size of application.

(4) Deployment

It is important to reduce production cost by mass production with common smart card platform. So NICSS standardized the first draft specification including the previously mentioned cost sharing model etc in April 2001 on the basis of our proposal. The model is called as NICSS framework.

The Ministry of Economy, Trade and Industry (METI) of Japan started IT CITY project based on the NICSS framework from January 2002 to March 2002. Based on NiNa model, NTT have individually developed the commercial use product called NICE and the product is deployed into the project and used in many areas in Japan. The scale of the project is as follows. The numbers of cards which were distributed are about 1.2 millions. The numbers of areas are 21. 100types/versions of applications were supported.

6.3 Discussion

(1) Utility

The survey results revealed that the function of application download is highly demanded, but with regard to using the general-purpose multi-application smart card to establish new card businesses, there are still some obstacles for business sectors to tackle with for the smooth transition. Finance and administrative (medical in particular) sectors are very serious and sensitive to information management, and without resolving and securing responsibility and security issues and unless the users gradually start noticing the usability, it may take about 4 years before this architecture takes root. It shows the progressiveness of our proposal also.

But the high evaluation of the NiNa technology, that uses licensing and policy control to provide a mechanism of sharing card operation cost, indicates the possibilities of establishing new card business by flexible communications between CI and SP within the same business sector. Take finance as an example, this NiNa technology will be a very effective tool to expand their original services and to enclose applications and users of their business partners. Receiving high evaluation on network adaptability shows that the usability is in high demand and SIM card in neutral business domain will be emerged horizontally and abruptly and be expanded into multiple business domains.

And there is requirement to provide with tailored pattern of policies for various business domains. So we need to prepare various types of policies for each business domain including cost-sharing pattern. Then we can improve or expand this platform to more effective end-user oriented one. In addition, federated nomadic cities as we mentioned in Sect. 2 can be realized by using this model.

At last as the application development environment, NiNa prototype adopted the JavaCard environment. It gave the following feature; 1) Portability; the cards have a standard internal API, which allows applications for one type of card to also be used on other types of JavaCards, 2) Java Applets; various Java development tools are available to support application development [11], [13]. So it provides not only good utility for CI/SP entities but also the prominent utility of JavaCard technology for application developers.

(2) Practicability

To put this technology into practical use, improving the performance is an issue especially for card processing. Actually in the prototype system we implemented security function using Java code. Then in the commercial version of system we confirmed that if we implemented it as Native code, we can dynamically reduce the time. The commercial product has been deployed at big real project and is running on well. So this platform can be provided with as practical use.

(3) Applicability

As mentioned above, IT enabled city project of Japanese government accepted the model based on our proposal. This shows the applicability of the proposal to the administrative or governmental sector as well as private sectors, because smart card applications of IT enabled cities include private sectors such as service points and digital money.

7. Related Technologies

Except for the previously mentioned major two approaches such as MULTOS and OP, there are many researches has been proposed. The Card Management System (CMS) and Application Management System (AMS) for multi-applications are now having important roles to control evolving APs on huge number of smart cards. Many internet based smart card researches has been proposed. They treated smart cards as a server.

Guthery et al. [19] proposed the way to treat Java card as a mobile Web server. Rees et al. [20] proposed a WebCard can be seen as an internet node.

Vandewalle et al. proposed JC-RMI to give the remote object invocation interface for applets on a smart card. The

GemXpresso RAD tool [21] generates a Java Card proxy from the card applet interface, and the proxy communicates with the card applet with APDUs. Thus the client AP can use the proxy as the card applet.

Rohs et al. [22] proposed the JiniCard provides smartcard middleware to retrieve smart card services over the internet. When a JiniCard is plugged somewhere, the JiniCard explore determines the capabilities of the card, then Jini registers the card and provides lookup services through Java interfaces. As its services are automatically registered, the JiniCard become available over the network thanks to Jini services.

Lorphelin [23] proposed the smartX framework to download new smart card APs on various terminals. The smartX engine is installed on the target terminals. The smartX applications are described in SML (Smart Markup Language). For example, SML provides ⟨Apdu⟩ tag to describe APDU command. SmartX terminal applications are developed by two blocks: the AP process and the AP protocol. The AP protocol is described in the SML dictionary and is card-specific. The AP process encapsulates the AP logic and compiled into Java applet. As the card-specific block is downloaded if necessary, this can minimize the AP downloading time.

Chan et al. [24] proposed the Java Card Web Servlet (JCWS) to provide a seamless access interface between a Web browser and a JavaCard. The JavaCard is viewed as a repository of Web-enabled object, HTML pages, data objects, and JavaCard Applets.

Urien et al. [25] proposed an internet smart card, @Card, works as an internet node including a web server and a trusted proxy. @Card has been implemented in a JavaCard and runs internet client and server AP. They also developed SmartTP looks like a TCP and connect smart agents located in both smart cards and terminals.

Bergner et al. [26] proposed a mechanism for connecting small devices to CORBA services. The architecture consists of smart card event broker on a smart card and a smartcard proxy in a terminal. The smart card proxy includes a proxy event broker and applet proxies. The smart card event broker and the proxy event broker communicate each other.

Urien [27] have realized an experimental XML script parser on smart cards. The XML script parser is invoked from an embedded web server. It can have access to all embedded resource4s and manage connections to remote servers.

Donsez et al. [28] proposed a JMS-SOAP based platform to connect multi-AP smart cards, and both to discover smart card services and requests for services. JMS is a Java based specification of Message Oriented Middleware (MOM). A SOAP proxy provides the facilities that allow distributed clients to discover and use the services on a smart card.

These approaches and targets differ from ours in that they provide the middleware connecting applications in the distributed environment and smart cards. The target of our platform is for downloading smart card applications to smart card in distributed environment, so it is also possible to combine these approaches with ours. Because downloading applications and connecting/executing applications are complementary technologies. For example, our card manager in Fig. 4 can be extended to have these distributed facilities as common facilities. Terminal and servers which support our platform can support those functionalities apart from our platform. Moreover by applying the above mentioned terminal technology such as SmartX and SML, we can easily and openly extend the diversification of terminal in combination with CCM (adapter).

8. Conclusion

In this paper, we introduced a smart card information platform 'NiNa', co-developed by NTT and a French company of Gemplus in October 2000, to investigate the ways and feasibility study of the implementation, functional and performance evaluations of the prototype for analyze the utility and practicability.

With licensing and policy management as the core technology, we accomplished a mechanism of CI-SP separated model, network-based flexible card operation post issuing, cost sharing, and adapter-based control and online download at the prototype. We have received excellent responses in the demonstrations as the first prototype achieved secure online download in a CI-SP separated environment.

From some investigations and deployment results, we confirmed that the concept of the NiNa technology is highly evaluated and the technology is applicable to the next-generation smart card platform.

Through these field trials, we will continue to discuss and evaluate the applications to the administrative sector, and establish a framework of offering the applications to a wide rage of business fields like medical, transport, communications and e-commerce.

Some issues relating to the multi-application smart card technology remain to be tackled. Future technology development will allow us to keep multiple applications in one card, but this also means that the more information one card can hold, the more complicated things become when we come to submit the various paperwork required to have the card re-issued if the card is lost, stolen or damaged. To cope with this situation, we must either keep multiple smart cards with the same authorities and information, or we must have a one-stop platform service agent on the server that can re-issue applications and restore the applications to their previous status prior to losing the card. We expect smart cards will be able to communicate with various other types of device such as mobile phones, IC card public pay phones, cars, ATMs, television sets, and game machines. We also require platform technology that can connect between these devices flexibly if we are to fully utilize the possibilities of these backyard information systems. Various kinds of platform and OS products have been developed for smart cards, and we believe the technology for connecting different smart card platforms will be developed and standardized further.

Finally we hope we can contribute to harmonize many international standardization organizations for this PKI-based card application downloading framework. And we expect this schema realize global nomadic worlds in which people can join (downloading applications) & use any types of services in any communities freely and trustingly on their demand.

Acknowledgements

First of all, we wish to express our gratitude to all the staff members of Gemplus, headed by Mr. Laurent den HOLLANDER and Mr. Eric CLEMENT for this productive cooperative research and development on the NiNa project.

We would like to thank all the people who is/was in NTT Laboratories as follows; Thanks should be given to Mr. Yasuhiro HOSODA, Mr. Masakazu KANBE for their coordination of application development. We would like to give our thanks to Mr. Shinichi YOSHIDA, Mr. Katsuhiko SUZUKI and Mr. Yuji HIURA for their development of Sapphire and evaluation of performance. We would like to appreciate Mr. Tadao TAKEDA and Mr. Masahiro YOSHIZAWA for their development of ELWISE card and collaboration with us.

And we would like to express our appreciation to members of NICSS for their collaboration.

Lastly, we wish to thank Mr. Seiichi IDO, who was the Executive Director of NTT Information Sharing Laboratory Group.

References

[1] NTT&Gemplus Collaborative project report, http://www.ntt.co.jp/news/news99e/9910/991013.html
[2] NICSS, http://www.nicss.gr.jp/main.htm
[3] IT CITY, http://www.itcity.jp
[4] MULTOS, http://www.multos.com
[5] OP(VOP), http://www.visa.com/nt/suppliers/open/main.html
[6] GP(OP), http://www.globalplatform.org
[7] JavaCard, http://www.javasoft.com/products/javacard/index.html
[8] S. Yamamoto, "Multi-application smart card platform—The way to the networked society," NTT Review, vol.14, no.1, pp.4–7, Jan. 2002.
[9] E. Niwano, H. Akashika, J. Hashimoto, and S. Yamamoto, "NiNa—A prototype of service oriented smart card platform," NTT Review, vol.14, no.1, pp.8–12, Jan. 2002.
[10] 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, Jan. 2002.
[11] K. Suzuki, S. Hirata, and S. Yamamoto, "A JavaCard VM for a 1-MB flash memory smart card," NTT Review, vol.14, no.1, pp.20–22, Jan. 2002.
[12] M. Yoshizawaq, H. Unno, T. Fukunaga, and H. Ban, "ELWISE—A super mutl-purpose smart card," NTT Review, vol.16, no.1, pp.23–27, Jan. 2002.
[13] E. Niwano, H. Akashika, K. Suzuki, S. Yoshida, S. Senda, and S. Yamamoto, "Sapphire on ELWISE taking Java card[tm] technology to a new level," JavaOne 2001, BUS-2492.
[14] R. Toji, Y. Wada, S. Hirata, and K. Suzuki, "A network-based platform for multi-application smart cards," Proc. Fifth IEEE International Conference (EDOC2001), pp.34–45, 2001.
[15] GemXpresso, http://www.conmweb.net/gemx-211.htm
[16] E. Niwano, S. Senda, S. Fujiwara, K. Okamoto, M. Teramoto, Y. Hosoda, and Y. Kokubun, "Concepts and design of CASA: A real-world-oriented distributes system architecture," The Fourth IFIP Conference on Intelligent Networks and Networks Intelligence, Feb. 1999.
[17] E. Niwano, K. Suzuki, N. Chiba, and Y. Hosoda, "Authority-oriented secure multi-application smart card OS—WAOS," IEICE Technical Report, KBSE, vol.99, no.254, pp.25–32, Aug. 1999.
[18] M. Algerganti, "Our dream of making the smart card into a genuine computer," Le Monde, 16th Nov. 2000.
[19] S. Guthery, R. Kehr, and J. Posegga, "How to turn GSM SIM into a web server," in 4th IFIP TC8/WG8.8 Working Conference on Smart Card Research and Applications, ed. J. Dommingo-Ferrer, D. Chan, and A. Watson, pp.209–222, 2000.
[20] J. Rees and P. Honneyman, "Webcard: A Java card Web server," in 4th IFIP TC8/WG8.8 Working Conference on Smart Card Research and Applications, ed. J. Dommingo-Ferrer, D. Chan, and A. Watson, pp.209–222, 2000.
[21] J. Vandewalle and E. Vetillard, "Developing smart card-based applications using Java card," Third Smart Card Research and Advanced Application Conference (CARDIS '98), pp.105–124, Sept. 1998.
[22] M. Rohs, H. Vogt, and R. Kehr, Plastic goes Internet: Issues in Smartcard Middleware, JavaCard Workshop, Cannes, 2000.
[23] X. Lorphelin, Internet and Smart Card Application Deployment, http://www.smartx.com
[24] A. Chan, J. Cao, H. Chan, and G. Young, "A web-enabled framework for smart card application in health services," CACM, vol.44, no.9, pp.77–82, Sept. 2001.
[25] P. Urien, H. Saleh, and A. Trizraoui, Internet Card, a smart card for Internet, http://proms2000.kt.agh.edu.pl, Protocols for Multimedia Systems, 2000.
[26] K. Bergner, A. Rausch, M. Sihling, and C. Vilsmeier, "CORBA and the Java card—Connecting small devices to a standard event service," 1st Symposium on Reusable Architectures and Components for Developing Distributed Information Systems (RACDIS '99), vol.I, pp.665–670, Aug. 1999.
[27] P. Urien, "Programming Internet smartcard with XML scripts," in E-smart 2001, ed. I. Attali and T. Jensen, LNCS 2140, pp.228–241, 2001.
[28] D. Donsez, S. Jean, S. Lecomte, and O. Thomas, "Turning multi-applications smart cards services available from anywhere at anytime: A SOAP/MOM approach in the context of Java cards," in E-smart 2001, ed. I. Attali and T. Jensen, LNCS 2140, pp.83–94, 2001.
[29] E. Niwano, M. Hatanaka, J. Hashimoto, and S. Yamamoto, "Early experience of a dynamic application downloading platform for multi-application smart cards," Proc. JCKBSE 2002.
[30] ISO/IEC 7816-4, 8, http://www.iso.ch/iso/en/ISOOnline.frontpage
[31] Open Card Framework - General Information Web Document, Open Card Consortium, Second Ed., Oct. 1998, http://www.opencard.org/
[32] E. Niwano, F. Duran, G. Lorenzo, and M. Faher, "Part4: MAS prerequisite: Core cross-sectoral architecture for interoperable multi-application systems," Multi-applications, vol.5, Open Smart Card Infrastructure for Europe (OSCIE) v2, eEurope Smart Card (eESC) Charter, March 2003.
[33] eESC, http://www.eeurope-smartcards.org/

Eikazu Niwano got the B.S. and M.S. in mathematics from Waseda University, Japan in 1987 and 1989 respectively. After he joined NTT Corporation in 1989, he has been engaged in research on distributed system architecture in the area of mobile multimedia messaging, agend computing, smart card computing and ubiquitous computing in this order. He is also a member of smart card-related National/European/International standardization organization now.

Junko Hashimoto received the B.S. and M.S. in information engineering from Kyushu University. She joined NTT in 1999. She is involved in reserch and development of Multi-application smart card platform.

Shoichi Senda completed his Master's Degree of Mathematics, began his career in NTT Laboratories in 1978. His research activities covered many areas such as telecommunication protocol, formal description technique, messaging system, directory system, and agent system. Since 1985 he has been involved in the standardization activities in CCITT (ITU) and joined the groups of X.400 Message Handling Systems and X.500 The Directory Recommendations from the very beginning. From 1988 to 1991 he was the special rapporteur of CCITT SG I Q.16 International Directory Services. He led the development team for prototyping X.500 directory systems from 1991. He joined the project developing public agent communication systems in 1994. In 1997, he was on a national project in confirming feasibility of Internet transferable electronic. Since then, he has been concentrating on global promotion of NTT's advanced technologies. And he has been a board director of GlobalPlatform since 2001.

Shuichiro Yamamoto joined NTT in 1979 and moved NTT DATA in 2002. He received an M.S. and D.R. in information engineering from Nagoya University. He is currently responsible for the research projects on grid computing, mobile computing, ubiquitous computing, service oriented computing, @business intelligence, and software engineering.

Masayuki Hatanaka joined NTT Labs, and was involved in research and development of a DIPS operating system (communication control) and CTRON-OS in 1978. In 1997, he was involved in R&D for an E-Money system, a Smart card platform, and a development promotion about an application platform. He has been the executive director of NTT Information Sharing Platform Laboratories since 2003.