APN UNIVERSAL PAYMENT GATEWAY

Dragomir D. Dimitrijević
Advanced Payment Networks, Ltd.

I INTRODUCTION

Advanced Payment Networks Ltd. (APN) is a privately owned enterprise located in Belgrade. It provides a wide range of consulting/engineering services in the area of non-cash payment and card technologies, including the development of its own card processing systems.

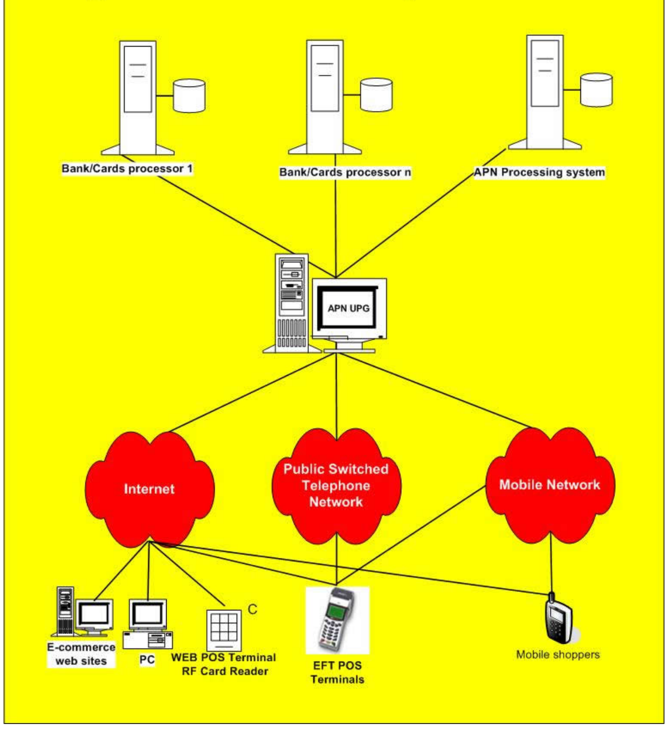
APN is the owner of Akademik Card™. It is the first, and so far unique product and payment identification system in the market of Serbia and Montenegro, specially created and intended for use in the academic environment. Presently, only Akademik Card is the banking payment card used for domestic payments via the Internet or mobile phones.

APN is proud to have on board members of the team which designed and implemented the first Yugoslav system for processing transactions on the Internet [1]. This highly successful, award winning system [2], operated by E-Bank, the first Yugoslav transaction processing company was in use between September 1999 and February 2002 when banks which used its services were closed. In this period, the system successfully sustained all of hundreds of attacks by hackers. In the Fall of 2004, founders of APN decided to revive the project applying new technologies to previously envisaged and in practice proven concepts. The newly developed system is fully compliant with the ISO 8583 standard for financial transactions. In this paper, we shall briefly describe the overall architecture of the developed system which implements the universal-commerce concept, i.e., allows commerce by all available communications means. Then we shall concentrate on the centrepiece of the system, the Universal Payment Gateway (UPG) and some of its functions. In particular, we shall describe how the UPG can be used to top-up prepaid user accounts such as prepaid mobile phones. More about other components of theAPN universal-commerce system may be found in [3].

II SYSTEM ARCHITECTURE

The universal-commerce (u-commerce) system architecture is shown in Figure 1. The adjective "universal" pertains to all available communications means, and not just to usual electronic (e-commerce) or mobile (m-commerce). Also "universal" pertains not just to financial transactions in the usual sense (credit/debit card payments), but also to custom made transactions such as loyalty cards, ID access control, contactless cards, etc.

Figure 1: Universal-Commerce System Architecture



The centerpiece of the system is the Universal Payment Gateway (UPG). It handles requests coming by various communications means. Requests for authorization of transactions made at various e-commerce sites are coming from the Internet using the TCP/IP protocol. Requests for transactions made face-to-face are coming from EFTPOS (Electronic Fund Transfer Point Of Sale) terminals via the public switched telephone networks. Throughout this document, we shall use shorter and widely adopted colloquial term POS. Requests may also come from mobile users in the form of SMS messages, from wireless POS terminals or WEB POS terminals (a special WEB based RF readers).

Based on their contents and types, requests are routed by the UPG toward various card processors. In the most traditional sense, payment requests are routed based on the card's BIN (Bank Identification Number) toward the appropriate card processor. In a less traditional sense, requests may be forwarded to custom made applications that handle requests. As an example, such applications may control access to premises and accordingly keep track of timesheets, or may keep track of points awarded to customers in a loyalty scheme. In addition, payment requests may also be handled by the UPG based on contents of reserved fields (national or private) in ISO 8583 messages. As it will be described in the next section, those fields may contain information about prepaid accounts (e.g., prepaid mobile phones) that have to be topped-up.

The other important part of the system architecture is the Virtual Point Of Sale (VPOS) terminal embedded in web e-commerce applications. Its role is to collect information needed for transaction authorization, format it according to the ISO 8583 standard, forward it toward the UPG, and receive the response.

In order to facilitate the introduction of new e-commerce businesses, APN has developed a number of e-commerce stores as out-of-box turnkey solutions. Such stores, which may be easily modified from a visual point of view, will satisfy most businesses from a functional point of view. The developed software may be used as building blocks for more specific business models and APN will be more than happy to assist such customers.

The APN software is developed using Java™ programming language[4], a de facto standard language for writing Internet applications. In particular, portability of Java and its "write once, run everywhere" concept, allow execution of the developed software on the widest range of computer platforms.

The low level transaction data exchange in accordance with the ISO 8583 standards relies on the JPOS library [5] which is a de facto standard for implementation of ISO 8583 software in Java. Some software in this library was used as-is while other had to be modified with respect to improved performance and accuracy. It is interesting to notice that this library still does not recognize the latest version of Dinar (ISO code 891 and name CSD) which will be done, per APN's intervention, in the next release of JPOS.

On the higher, application level, we use proprietary API (Application Programming Interface) to execute high level functions such as credit or payment. In the future, we plan to use the Java's JPAY API when it becomes standardized in the Java Community Process.

Although the APN code may run on virtually any operating system and database, due requirements for high throughput, reliability, and availability of the UPG, we decided to run our software on Linux™ operating system and Postgresql™ database. Internet stores may run without any problems on systems such as any version of Microsoft Windows™ operating system and low-end databases such as Microsoft Access™ too.

As a web server, we recommend Tomcat™, a derivative of Apache™ server[6]. The Apache server is the most widely used server with more than 50% of all installations of web servers in the world.

III UNIVERSAL PAYMENT GATEWAY

The role of UPG is to rout ISO 8583 messages toward card processors according to the predefined rules and obtain authorizations for requested financial transactions. In addition, the UPG may perform other functions such as topping-up prepaid accounts (e.g., prepaid mobile phones). Initially we used name IPG (Internet Payment Gateway) since it received requests from the Internet only. In a later development phase, we had to allow access from POS terminals via telephone lines. As the requirements grew, the IPG evolved to UPG indicating that it can accept requests from multiple types of communications channels. The structure of UPG is shown In Figure 2.

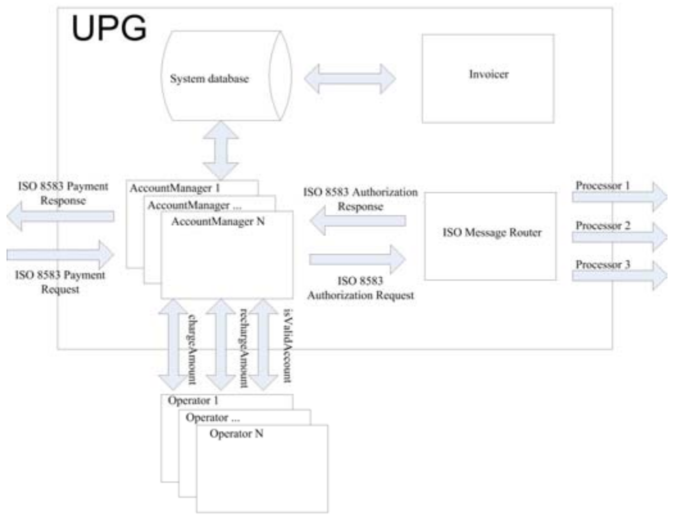


Figure 2: Universal Payment Gateway

The UPG is highly configurable using numerous parameters in its configuration file. . As far as input channels are concerned, it is possible to configure their type (Internet, telephone lines, or SMS), level of security protection, version of ISO 8583 messages, and level of logging of processed ISO messages with respect to their type (Message Type Indicator and Processing Code) and individual message fields (some fields may be stored and some not as defined in the configuration file).

Payment requests arrive to the UPG in the form of ISO 8583 messages as shown in the left-hand side of Figure 2. Each message is examined by an array of so called account managers. Each account manager corresponds to one of operators (e.g., mobile phone operators or Internet service providers). Account managers sequentially, one by one, examine ISO messages and try to recognize content intended for their operator. For example, an account manager for a prepaid mobile telephone operator looks for the appropriate area code in the phone number located in one of the reserved field of ISO messages. It is assumed that an account manager (and consequently the corresponding operator) may be uniquely identified based on the message content, i.e., no multiple account managers may think the message is intended for them. In case no account manager is identified, the message is handled as an ordinary payment request and simply forwarded to one of card processors.

The UPG sees operators through a Java interface called AccountManager which hides all details specific to individual account managers and their corresponding operators. In such a way, each account manager/operator is treated uniformly regardless to low level details such as format of adentifiers of accounts (e.g., phone number), the way account identifiers are embedded in ISO messages, communication protocol between UPG and operator, etc. Most of methods defined in the AccountManager interface are implemented by an abstract Java class called AbstractAccountManager. Only five methods in this class are defined as "abstract" and are implemented by subclasses of AbstractAccountManager. These methods that hide low level implementation details of specific account managers/operators are:

- Method getAccount tries to recognize and extract the account identifier (e.g., mobile phone number) from an ISO 8583 message. In case the message is not intended to that particular account manager/operator, a null value is returned. If all account managers return a null value, the request is treated as an ordinary payment.
- Method isValidAccount contacts the corresponding operator and verifies if the account is valid (e.g., if the phone number is valid and if it is a prepaid phone).
- Method reChargeAmount contacts the corresponding operator and credits the specified account with the specified amount of money/points.
- Method chargeAmount dontacts the corresponding operator and takes away the specified amount of money/points from the specified account.
- Method reverse reverses the specified transaction.

When an account manager recognizes an ISO message that is intended to its operator, i.e., when its method getAccount returns a non-null value, the account is verified in consultation with the operator using the isValidAccount method.

In order to authorize the financial transaction, the UPG forwards the request toward one of several card processors. Multiple processors may be used for:

- Functional reasons as they may process different card types;
- Higher reliability purposes as one processor may take over in case another one fails;
- Load balancing purposes as one of the card types may use one processor as the primary one while the primary processor of another card type may be used as a back-up and vice versa.

The communication channel toward each individual processor may be a real communication channel and the processor may be a remote computer. The channel and the processor may be also a local custom-made application which may, as an example, keep track of award points in a local database with a loyalty scheme data.

The UPG may be configured to recognize or ignore various card types based on their BIN. Each card type has a prioritized list of processors assigned. If the first processor on the list is not available (the defined timeout expires before the response is received) the next one is contacted and so on.

In case the financial transaction is authorized successfully by the card processor, the corresponding operator is contacted using reChargeAmount method, the account is credited wit the specified amount of money/points, and the customer is informed about success (or failure) of the transaction.

The information about transactions is stored in the system database. The database contains the business model defined including information about:

- Companies involved in the business model including their roles (UPG processor, distributors, agents, owners of POS terminals, and operators);
- POS terminals in the network including their statuses (active/disabled) and companies related to them (owner, agent, distributor, and operators they are allowed to serve);
- Contracts that define their start/ending dates, companies involved and the way proceeds from sale are divided among them.

A process called "Invoicer" runs in the system in the background. It periodically at specified times calculates total sale in the invoicing period,andproceeds that belong to each company as defined by the currently active contract.

IV IMPLEMENTATIONAL ISSUES

As it was mentioned earlier, the UPG is highly configurable. Here, we shall describe how account managers may be configured, while keeping their implementational details totally isolated from the rest of the UPG code. One of the methods in the AccountManager interface is setConfiguration, whose role is to initialise/configure the account manager. This method is implemented by the AbstractAccountManager class and may be overwritten by its subclass in case initialisation of additional parameters is needed as follows:

```
public void setConfiguration(
  Configuration cfg )
   throws ConfigurationException
{
   // Common configuration
   super.setConfiguration( cfg );
   // Code specific for this account
// manager …
}
```

In the above code, Configuration is the interface to a configuration file such as XML or Java properties file. Similarly, the UPG is configured using its own setConfiguration method and configuration file. Within this method, account managers are configured using code similar (excluding appropriate try/catch clauses) to the following:

```
String cls = cfg.get("class");
AccountManager am =
  Class.forName(cls).newInstance();
am.setConfiguration( cfg );
```

The first line in the above code extracts from the UPG configuration file the full class name (including both class and package name) of the class that implements the particular account manager. The second line dynamically loads and instantiates the account manager using its class name and the corresponding default (no-args) constructor. The third line configures the account manager using the setConfiguration method and parameters specified in the configuration file.

The described technique totally isolates implementation of the account manager from the rest of UPG code. A developer of an account manager does not need to know anything about implementation of the UPG. The implementation of a new account manager (and consequently introduction of a new operator) is reduced to several pages of code which can be plugged-in the UPG by adding appropriate parameters in the configuration file. Similar technique is used with other elements of the APN universal commerce system. They can be configured/built using separately developed modules as building blocks.

The APN universal commerce system consists of several products: the UPG, the VPOS, and a number of out-of-box Internet stores. The project is still under development and growing. Just to illustrate its complexity we should say that, at present time, it consists of 7 databases, more than 250 files of which more than 180 are Java classes and resource files.

V CONCLUSION

In this paper we described the APN Universal Commerce System and in particular its centrepiece the Universal Payment Gateway. One of UPG's functions, topping-up prepaid accounts was described and some implementational issues were discussed.

REFERENCES

[1] D. Dimitrijević, T. Apostolska, A. Vujić, and P. Nikolić, "E-Bank: Prvi jugoslovenski sajt za elektronsku trgovinu sa kompletnom uslugom" (E-Bank: The first Yugoslav full-service e-commerce site), JISA Info, Number 6, 1999, pp. 25-27.

[2] Diskobolos 2000 in the finance category - an annual award granted by the Yugoslav Informatic Alliance to the Virtual POS Terminal software, http://www.jisa.org.yu/2000.htm .

[3] D. Dimitrijević, "APN Family of Universal-Commerce Products," E-Trgovina Conference 2005, Palić, April 2005.

[4] Java home page, http://java.sun.com.

[5] JPOS home page, http://www.jpos.org.

[6] Apache home page, http://www.apache.org.

Abstract: As the market leader in Serbia and Montenegro in the area of non-cash payment and card technologies, Advanced Payment Networks Ltd. has developed a family of universal-commerce products. The adjective "universal" pertains to multiple communications means in the system (Internet, as well as wireline and mobile telephone networks) and use of cards which goes well beyond traditional use as a means of non-cash payment. In this article we describe the overall system architecture as well as its centrepiece – the Universal Payment Gateway (UPG). As one of the UPG's functions, we described its ability to top-up prepaid accounts such as prepaid mobile phones.

APN UNIVERSAL PAYMENT GATEWAY, Dragomir D. Dimitrijević.

Advanced Payment Networks, Ltd.
Prote Mateje 56
11000 Belgrade
Serbia and Montenegro
Phone/Fax: +381-11-444-7272, +381-11-344-4716
E-mail:dragomir.dimitrijevic@apnet.co.yu
WWW: http://www.apnet.co.yu