Network Working Group
Request for Comments: 3506
Category: Informational

K. Fujimura
NTT
D. Eastlake
Motorola

March 2003

Requirements and Design for Voucher Trading System (VTS)

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

Crediting loyalty points and collecting digital coupons or gift certificates are common functions in purchasing and trading transactions. These activities can be generalized using the concept of a "voucher", which is a digital representation of the right to claim goods or services. This document presents a Voucher Trading System (VTS) that circulates vouchers securely and its terminology; it lists design principles and requirements for VTS and the Generic Voucher Language (GVL), with which diverse types of vouchers can be described.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Table of Contents

1. Background

It is often necessary to credit loyalty points, collect digital coupons or gift certificates, etc, to complete purchases or other trading transactions in the real world. The importance of these activities is also being recognized in Internet Commerce. If a different issuing or collecting system to handle such points or coupons must be developed for each individual application, the implementation cost will be excessive, inhibiting the use of such mechanisms in electronic commerce. Consumers may also be forced to install a number of software modules to handle these points or coupons.

A voucher is a digital representation of the right to claim services or goods. Using vouchers, a wide-range of electronic-values, including points or coupons, can be handled in a uniform manner with one trading software module.

This document presents the terminology and model for a Voucher Trading System (VTS) that circulates vouchers securely; it also lists design principles and requirements for a VTS and the Generic Voucher Language (GVL), with which diverse types of vouchers can be described.

2. Terminology and Model

2.1 Voucher

A voucher is a digital representation of the right to claim goods or services. To clarify the difference between vouchers and electronic money/digital certificates, we introduce a formal definition of vouchers in this document.

> Let I be a voucher issuer, H be a voucher holder, P be the issuer's promise to the voucher holder. A voucher is defined as the 3-tuple of <I, P, H>.

Examples of P are as follows:

- Two loyalty points are added to the card per purchase. If you collect 50 points, you'll get one item free. (Loyalty points)
- Take 10% off your total purchase by presenting this card. (Membership card)
- Take 50% off your total purchase with this coupon. The purchase transaction uses up the coupon. (Coupon)
- The bearer can access "http://..." for one month free. (Free ticket for sales promotion)
- The bearer can exchange this ticket for the ordered clothes. (Exchange ticket or Delivery note)
- Seat number A-24 has been reserved for "a-concert" on April 2. (Event ticket)

Note that P does not need to be described in terms of a natural language as long as the contents of the vouchers are specified. For example, a set of attribute name and value pairs described in XML can be employed to define the contents.

2.2 Participants

There are four types of participants in the voucher trading model: issuer, holder, collector, and VTS provider. Their roles are as follows:

Issuer: Creates and issues a voucher. Guarantees contents of the voucher.

Holder (or user): Owns the vouchers. Transfers and redeems the voucher to other users or collector.

Collector (or examiner): Collects or examines the voucher and implements its promise. In general, compensated by goods or services rendered.

VTS Provider: Provides a VTS and guarantees that a particular voucher is not assigned to multiple holders or used multiple times unless permitted for that voucher type.

The IOTP model [IOTP] includes merchant, deliverer, consumer and other participants. They take various roles in the settlement because a merchant, for example, can be considered as an issuer, or holder depending on whether the merchant creates the voucher her/himself or purchases it from a wholesaler or manufacturer. A merchant can also be a collector if the shop collects gift certificate or coupons.

2.3 Voucher Trading System (VTS)

A voucher is generated by the issuer, traded among holders (users), and finally is collected by the collector:

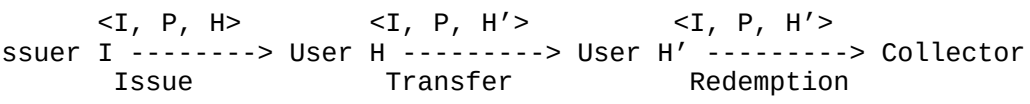

Figure 1. Life cycle of vouchers

The VTS provider supplies a VTS that enables vouchers to be circulated among the participants securely.

A formal definition of VTS is as follows:

> A voucher trading system (VTS) is a system that logically manages a set of valid vouchers VVS, which is a subset of {<I, P, H> | I in IS, P in PS, H in HS} where IS is the set of issuers, PS is the set of promises, and HS is the set of holders; VTS prevents them from being modified or reproduced except by the following three transactions: issue, transfer, and redemption. The initial state of the VVS is an empty set.
>
> Note that this does not imply that VVS is stored physically in a centralized database. For example, one implementation may store vouchers in distributed smart cards carried by each holder [T00],

or may store them in multiple servers managed by each issuer or trusted third parties. This is a trust policy and/or implementation issue [MF99].

Issue
An issue transaction is the action that creates the tuple of <I, P, H> and adds it to the VVS with the issuer's intention.

Transfer
A transfer transaction is the action that rewrites the tuple of <I, P, H> (in VVS) as <I, P, H'> (H<>H') to reflect the original holder H's intention.

Redemption
There are two redemption transactions: presentation and consumption.

A presentation transaction is the action that shows the tuple of <I, P, H> (in VVS) to reflect the holder H's intention. In this case, the ownership of the voucher is retained when the voucher is redeemed, e.g., redemption (presentation) of licenses or passports.

A consumption transaction is the action that deletes the tuple of <I, P, H> (in VVS) to reflect the holder H's intention and properties of the voucher. The ownership of the voucher may be voided or the number of times it is valid reduced when the voucher is redeemed, e.g., redemption of event tickets or telephone cards.

Note that one or more of these transactions can be executed as part of the same IOTP purchase transaction. See details in Section 6.

3. VTS Requirements

A VTS must meet the following requirements

(1) It MUST handle diverse types of vouchers issued by different issuers.

(2) It MUST prevent illegal acts such as alteration, forgery, and reproduction, and ensure privacy.

(3) It MUST be practical in terms of implementation/operation cost and efficiency.

Each of these requirements is discussed below in detail.

3.1 Capability of handling diversity

(a) Different issuers

Unlike a digital cash system that handles only the currency issued by a specific issuer such as a central bank, the voucher trading system MUST handle vouchers issued by multiple issuers.

(b) Various types of vouchers

Unlike a digital cash system that only handles a currency, the system MUST handle various types of vouchers, such as gift certificates, coupons, and loyalty points.

3.2 Ensuring security

(c) Preventing forgery

Only the issuer can cause a valid voucher to be issued. It MUST NOT be possible for other parties to cause a valid voucher to be created.

(d) Preventing alteration

Voucher MUST NOT be altered during circulation except that the transfer transaction, in which the voucher holder is rewritten, is permitted. Only the current holder can initiate a transfer transaction.

(e) Preventing duplicate-redemption

A voucher MUST NOT be redeemable once it has been consumed (the result of some redemption transactions). Only the holder can initiate a redemption transaction.

(f) Preventing reproduction

Voucher MUST NOT be reproduced while in circulation. That is, there must be only one valid holder of any particular voucher at any particular time.

(g) Non-repudiation

It SHOULD NOT be possible to the issuer to repudiate the issuance, or the holder to repudiate the transfer or redemption of a voucher, after it is issued, transferred or redeemed.

(h) Ensuring privacy

Current and previous holders of a voucher SHOULD be concealed from someone coming into possession of the voucher.

(i) Trust manageability

If a wide variety of vouchers are in circulation, it might be difficult for users to judge whether a voucher can be trusted or not. To assist such users, a trust management function that verifies the authenticity of a voucher SHOULD be supported.

3.3 Ensuring practicality

(j) Scalability

A single centralized broker that sells all types of vouchers, or a centralized authority that authenticates all issuers or other participants, SHOULD NOT be assumed. A system that relies on a single centralized organization is excessively frail; failure in that organization causes complete system failure.

(k) Efficiency

It MUST be possible to implement VTS efficiently. Many applications of vouchers, e.g., event ticket or transport passes, require high performance, especially when the voucher is redeemed.

(l) Simplicity

It SHOULD be possible to implement VTS simply. Simplicity is important to reduce the cost of implementation. It is also important in understanding the system, which is necessary for trust in the system.

4. Scope of VTS Specifications

To implement a VTS, Voucher Trading Protocol (VTP), VTS Application Programming Interface (VTS-API), and Generic Voucher Language (GVL) must be developed. The objectives, benefits, and limitations of standardization for each specification are discussed below.

4.1 Voucher Trading Protocol

To achieve interoperability among multiple VTSs developed by independent VTS Providers, standard protocols for issuing, transferring, or redeeming vouchers will be needed. However, there are several ways of implementing VTS. For discount coupons or event

tickets, for example, the smart-card-based decentralized offline VTS is often preferred, whereas for bonds or securities, the centralized online VTS may be preferred. It is impractical to define any standard protocol at this moment.

4.2 VTS-API

To provide freedom in terms of VTS selection for issuers and application developers, a standard Voucher Trading System Application Programming Interface (VTS-API) that can encapsulate VTS implementations should be specified. It allows a caller application to issue, transfer, and redeem voucher in a uniform manner independent of the VTS implementation. Basic functions, i.e., issue, transfer, and redeem, provided by VTS-API can be straightforwardly derived from the VTS model described in this document. More design details of the VTS-API will be discussed in a separate document or a separate VTS-API specification.

4.3 Generic Voucher Language

To satisfy the diverse requirements placed on VTS (see Section 3), a standard Generic Voucher Language (GVL) that realizes various voucher properties should be specified. This approach ensures that VTS is application independent. The language should be able to define diverse Promises P of the voucher <I, P, H> to cover tickets, coupons, loyalty points, and gift certificates uniformly. Specifying I and H is a VTS implementation issue and can be achieved by using a public key, hash of a public key, URI or other names with scope rule.

In the following section, we discuss GVL Requirements in detail.

5. GVL Requirements

5.1 Semantics

Semantics supported by the language and their requirements levels are described below in detail.

(a) Validity control

The invalidation (punching) method that is executed when the voucher is redeemed depends on the type of the voucher. For example, a loyalty point will be invalidated if the point is redeemed but a membership card can be used repeatedly regardless of the number of times presented. The language MUST be able to define how validity is modified. Additionally, the language MUST be able to define the validity period, start date and end date.

(b) Transferability control

Some types of vouchers require transferability. The language MUST be able to specify if a voucher can be transferred.

(c) Circulation control

Depending on the type of the voucher, various circulation requirements or restrictions must be satisfied [F99], for example, only qualified shops can issue particular vouchers or only a certain service provider can punch (invalidate) particular vouchers. The language SHOULD be able to specify such circulation requirements.

(d) Anonymity control

Different types of voucher will require different levels of anonymity. The language SHOULD be able to achieve the required level of anonymity.

(e) Understandability

The terms and description of a voucher SHOULD be objectively understood by the participants, because this will contribute to reducing the number of disputes on the interpretation of the vouchers promised.

(f) State manageability

Some types of vouchers have properties the values of which may change dynamically while in circulation, e.g., payment status, reservation status, or approval status. The language MAY support the definition of such properties.

(g) Composability

Some types of vouchers consist of several sub-vouchers, which may be issued separately from the original vouchers typically because the vouchers are issued by different organizations or issued at different times. The language MAY support compound vouchers composed of multiple sub-vouchers.

5.2 Syntax

To achieve consistency with other related standards shown below, the syntax of the language MUST be based on XML [XML].

The language syntax MUST enable any application-specific property, e.g., seat number, flight number, etc. to be defined. A schema definition language that can be translated into application-specific DTDs may be needed.

5.3 Security

The language MUST provide the parameters necessary to establish security. Security requirements, however, mainly follow VTS requirements described in Section 3 rather than GVL requirements.

5.4 Efficiency

The vouchers may be stored in a smart card or PDA with a restricted amount of memory. Large definitions may incur long transfer and processing times, which may not be acceptable. The language SHOULD enable the efficient definition of vouchers

5.5 Coordination

The language specification SHOULD be consistent with the following specifications:

(1) Internet Open Trading Protocol v1.0 [IOTP]
(2) XML-Signature [XMLDSIG]
(3) Extensible Markup Language (XML) Recommendation [XML]
(4) ECML Version 2 [ECML]

5.6 Example of GVL

An example of a voucher definition in GVL is described below. This example defines a five dollar discount coupon for specific merchandise, a book with ISBN number 0071355014. This coupon is circulated using a VTS called "Voucher Exchanger". To claim this offer, one coupon must be spent. The coupon is valid from April 1st in 2001 to March 31st in 2002.

```
<?xml version="1.0"?>
<Voucher xmlns="urn:ietf:params:xml:schema:vts-lang"
         xmlns:vts="http://www.example.com/vts">
  <Title>IOTP Book Coupon</Title>
  <Description>$5 off IOTP Book</Description>
  <Provider name="Voucher Exchanger">
    <vts:Version>VE2.31</vts:Version>
  </Provider>
  <Value type="discount" spend="1">
    <Fixed amount="5" currency="USD"/>
  </Value>
```

```
  <Merchandise>
    <bk:Book xmlns:bk="http://www.example.com/bk"
             bk:isbn="0071355014"/>
  </Merchandise>
  <ValidPeriod start="2001-04-01" end="2002-03-31"/>
</Voucher>
```

6. Application Scenarios

This section describes, as a typical electronic commerce example involving advertisement, payment, and delivery transactions, the use of vouchers and VTS, and shows that vouchers can be used as an effective way to coordinate autonomous services that have not yet established trust among each other.

Figure 2 shows a typical electronic commerce example of a consumer searching for goods or services and making a purchase:

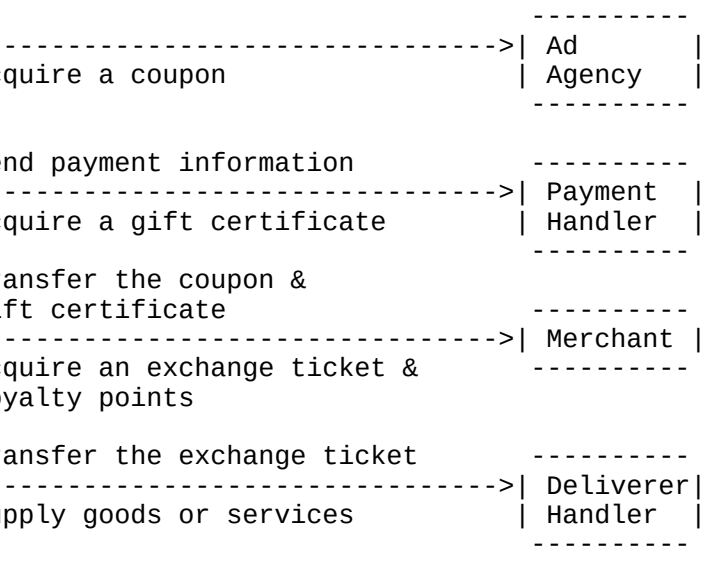

Figure 2. Application example of vouchers

(1) Use a search engine to find the desired goods or services and acquire a coupon from an ad agency that represents the right to purchase the goods or services at a discounted price.

(2) Acquire a gift certificate from a payment handler in exchange for cash or payment information.

(3) Transfer the coupon and gift certificate to the merchant, and in exchange acquire an exchange ticket and loyalty points.

(4) Transfer the exchange ticket to the deliverer handler and receive the goods or services.

In this example, the coupon, gift certificate, and exchange ticket each represent the media that yields the above four transactions.

Note that it is not necessary to trust the participants involved in the transactions, but to trust the vouchers themselves. In other words, there is no need to exchange contracts among the participants beforehand if the vouchers themselves are trusted.

Take the exchange ticket as an example; even if the delivery handler does not trust the consumer, the merchant that issued the exchange ticket is trusted, and if the VTS guarantees that there is no duplication in the trading process of the exchange ticket, there is no problem in swapping the exchange ticket for the goods or services. In the same way, even if the merchant does not trust the delivery handler, the issuance of the exchange ticket can be verified, and if the VTS guarantees that there is no duplication in the trading process of the exchange ticket, there is no problem in swapping the exchange ticket for the goods or services (Fig. 3). In other words, if there is trust in the issuer and the VTS, trust among the participants involved in the transactions is not required.

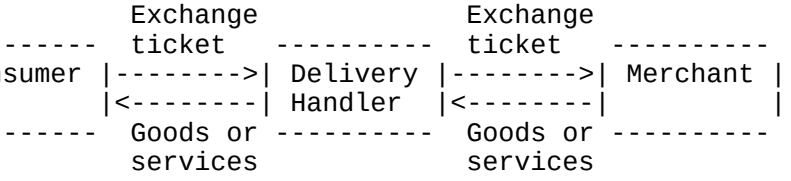

Figure 3. Coordination of untrusted participants using exchange ticket

In general, it is more difficult to trust individuals than companies, so this characteristic of VTS is especially important.

Moreover, the transactions involving vouchers have desirable features with respect to privacy protection. For example, in the above exchange ticket scenario, the consumer can designate the delivery service for himself, so the merchant does not even need to know any personal information such as the delivery address. Furthermore, by designating a convenience store etc. as the receiving point, the delivery service does not need to know the address of the consumer.

7. Q & A

- Is it possible to implement a VTS using digital certificates?

  If transferability is not required, a voucher can be easily implemented as a digital certificate, i.e., Signed_I(I, P, H), where the phrase "Signed_I" means that the entire block is signed by the issuer's digital signature. If transferability is required, then H is changed during the transfer, i.e., the signature is broken. Additionally, online data base checking or tamper-resistant devices are required to prevent duplicate-redemption.

- What is the difference from digital-cash?

  VTS must handle various types of vouchers, such as gift certificates, coupons, or loyalty points unlike a digital cash system which handles only currency. Additionally, vouchers are issued by different issuers.

- Is it possible to support "digital property rights?

  Digital property rights can be represented as a voucher and can be traded using VTS. However, some protected rendering system would be required to regenerate the digital contents securely in order to support digital property rights. These requirements are out of scope of VTS.

8. Security Considerations

Security issues are discussed in Section 3.2 and 5.3.

9. Acknowledgments

I would like to thank Masayuki Terada and Perry E. Metzger, for their valuable comments.

10. References

[ECML] ECML Version 2, Work in Progress.

[F99] K. Fujimura, H. Kuno, M. Terada, K. Matsuyama, Y. Mizuno, and J. Sekine, "Digital-Ticket-Controlled Digital Ticket Circulation", 8th USENIX Security Symposium, August 1999.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[IOTP] Burdett, D., "The Internet Open Trading Protocol", RFC 2801, April 2000.

[MF99] K. Matsuyama and K. Fujimura, "Distributed Digital-Ticket Management for Rights Trading System", 1st ACM Conferences on Electronic Commerce, November 1999.

[T00] M. Terada, H. Kuno, M. Hanadate, and K. Fujimura, "Copy Prevention Scheme for Rights Trading Infrastructure", 4th Smart Card Research and Advanced Application Conference (CARDIS 2000), September 2000.

[XML] "Extensible Mark Up Language (XML) 1.0 (Second Edition)", A W3C Recommendation, <http://www.w3.org/TR/REC-xml>, October 2000.

[XMLDSIG] "XML-Signature Syntax and Processing", A W3C Proposed Recommendation, <http://www.w3.org/TR/xmldsig-core>, August 2001.

11. Authors' Addresses

Ko Fujimura
NTT Corporation
1-1 Hikari-no-oka
Yokosuka-shi
Kanagawa, 239-0847 JAPAN

Phone: +81-(0)468-59-3814
Fax: +81-(0)468-59-8329
EMail: fujimura@isl.ntt.co.jp

Donald E. Eastlake 3rd
Motorola
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-851-8280
EMail: Donald.Eastlake@motorola.com

12. Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.