

Trade Working Group  
INTERNET-DRAFT

Expires: August 2001

February 2001  
Ko Fujimura  
Masayuki Terada  
NTT

XML Voucher: Generic Voucher Language  
<[draft-ietf-trade-voucher-lang-00.txt](http://draft-ietf-trade-voucher-lang-00.txt)>

## Status of This Document

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

Distribution of this document is unlimited. Please send comments to the TRADE working group at <ietf-trade@lists.elistx.com>, which may be joined by sending a message with subject "subscribe" to <ietf-trade-request@lists.elistx.com>.

Discussions of the TRADE working group are archived at  
<http://www.elistx.com/archives/ietf-trade>.

## Abstract

This document specifies rules for defining voucher properties in XML syntax. A voucher is a logical entity that represents a right to claim goods or services. A voucher can be used to transfer a wide-range of electronic-values, including coupons, tickets, loyalty points, and gift certificates, which are often necessary to process in the course of payment and/or delivery transactions.



## Table of Contents

|  |                   |
|--|-------------------|
| Status of this Memo .....                              | <a href="#">1</a> |
| Abstract .....   | <a href="#">1</a> |
| <a href="#">1</a> . Introduction .....                 | <a href="#">2</a> |
| <a href="#">2</a> . Processing Model .....             | <a href="#">2</a> |
| <a href="#">3</a> . Trust Model .....                  | <a href="#">3</a> |
| <a href="#">4</a> . Component Structure .....          | <a href="#">4</a> |
| <a href="#">4.1</a> Voucher Component .....            | <a href="#">4</a> |
| <a href="#">4.2</a> Promise Component .....            | <a href="#">4</a> |
| <a href="#">5</a> . Syntax Overview and Examples ..... | <a href="#">6</a> |
| <a href="#">6</a> . Semantics .....                    | <a href="#">7</a> |
| <a href="#">7</a> . DTD .....                          | <a href="#">7</a> |
| <a href="#">8</a> . Security Considerations .....      | <a href="#">7</a> |
| <a href="#">9</a> . Acknowledgments .....              | <a href="#">7</a> |
| <a href="#">10</a> . References .....                  | <a href="#">7</a> |
| <a href="#">11</a> . Author's Address .....            | <a href="#">8</a> |

## [1](#). Introduction

This document, XML Voucher, specifies rules for defining voucher properties in XML syntax. The motivation and background of the specification is described in [[GVT](#)].

A voucher is a logical entity that represents a certain right and logically managed by the Voucher Trading System (VTS). A voucher is generated by the issuer, and traded among users, and finally is collected by the collector using VTS.

This document defines syntax and semantics of the Voucher Component that is used to define voucher meaning and processing rules in XML syntax [[XML](#)]. In a Voucher Component, properties needed to allow the voucher to be processed by VTS or other trading systems, e.g., wallet or merchant system, are described. VTS definitions and models are also defined in [[GVT](#)].

Note: This document uses a "voucher" as an "instance of voucher" whose meaning is defined by Voucher Component. In other words, multiple vouchers can be issued and managed by the VTS using the same Voucher Component.

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 [[RFC 2119](#)]

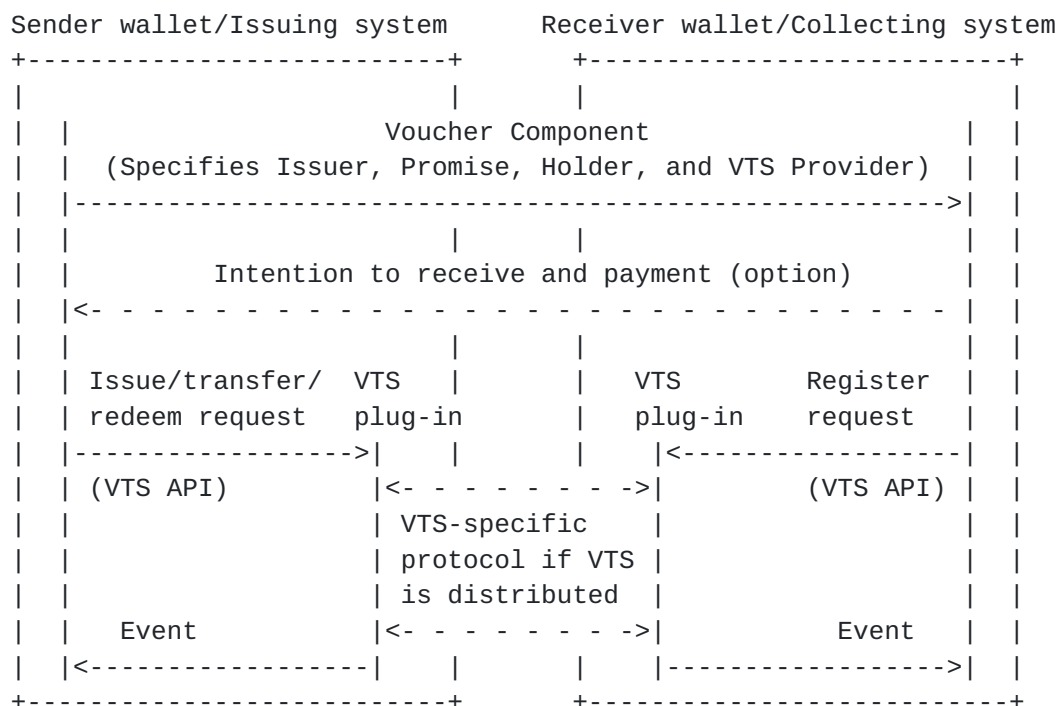
## **2. Processing Model**

There are several ways of implementing VTS and technologies are

K. Fujimura, M. Terada

[Page 2]

To provide implementation flexibility, this document assumes a modular wallet architecture that allows multiple VTS to be added as plug-ins. In the architecture, instead of specifying a standard voucher transfer protocol, two specifications, i.e., Voucher Component and VTS API specifications, are standardized (Figure 1).



After sender and receiver agree on what vouchers are to be traded and which VTS is to be used, the issuing system or wallet system requests the corresponding VTS plug-in to permit the issue, transfer, or redeem transactions to be performed via the VTS API. The VTS then rewrites the ownership of the vouchers using a VTS-specific protocol. Finally, a completion event is sent to the wallet systems or issuing/collecting systems.

A voucher is trusted if the issuer and VTS provider are trusted, since the issuer is responsible for the contents of the voucher and the VTS provider is responsible for preventing ownership from being

assigned to multiple users. This model enables trading partners to verify the trust of the voucher regardless of the trust of the partners.

The trust level required for issuer and VTS provider depends on the type (or Promise) of the voucher. To provide the information needed for the verification, the conditions of issuer and VTS provider are specified in the Voucher Component.

In this case, however, if a malicious user could alter the Voucher Component, a forged voucher, would be verified as valid. This document, therefore, assumes that such alteration is impossible during delivery of the Voucher Component; this is possible with existing technologies, such as [[XMLDSIG](#)] or [[TLS](#)].

Note: The Voucher Component does not have to be sent from the sender of the voucher. It can be directly delivered from the trusted issuer or trusted third party using TLS or other secure communication channel. Note also that a set of trusted Voucher Components can be pre-downloaded before conducting a transaction.

## **[4.](#) Component Structure**

### **[4.1](#) Voucher Component**

A Voucher Component provides VTS branding information, and basic properties for representing a voucher, i.e., issuer, promise, and holder. Implementation-specific properties are often required for authenticating issuer and holder. These implementation-specific properties of the VTS can be attached as child elements using [[XML-ns](#)].

The Voucher Component contains Provider Component, Issuer Component, Promise Component, and Holder Component as follows:

#### Provider Component

Provides properties to specify which VTS Provider (or VTS plug-in) can be used for trading the voucher.

#### Issuer Component

Provides properties specifying the issuer of the vouchers. This is optional and can be omitted if the issuer role is delegated to the VTS Provider.

#### Promise Component

Provides properties used by the application system of VTS, e.g., wallet system, merchant system. The Promise Component is transparent to the VTS and is described in [Section 4.2](#).

#### Holder Component

Provides properties to specify the holder of the vouchers. This is optional and can be omitted if the vouchers are transferable. (Note: Even for transferable vouchers, this



component may be used by the VTS depending on the implementation.)

## **4.2 Promise Component**

The Promise Component provides common properties useful for displaying and manipulating wallet systems. It includes monetary property (value) of the voucher. These monetary properties are needed to calculate the amount paid when the vouchers are redeemed at Merchant site, etc.

The Promise Component contains Title Component, Description Component, ValidPeriod Component, Redemption Component, Merchandise Component, and Value Component as follows:

### Title Component

Provides the title of the voucher. This is mainly for displaying the list of entities stored in a wallet system.

### Description Component

Provides a short description of the voucher. This is mainly for displaying the entities stored in a wallet system.

### ValidPeriod Component

Indicates voucher's validity period, start date and end date.

### Redemption Component

Provides the number of vouchers to be redeemed for claiming the merchandise or financial value specified in Merchandise Component or Value Component. If "n" (>0) is specified, the merchandize can be claimed in exchange with "n sheets of" vouchers. (Note: Multiple vouchers for the same Voucher Component must exist in this case.) If "0" is specified, the vouchers do not need to be consumed. It can be used repeatedly regardless of the number of times redeemed.

### Merchandise Component

Provides domain-specific meaning of the voucher, e.g., reference number of the merchandize or seat number for an event ticket, which is needed to identify the merchandize rendered when the voucher is redeemed. The properties of this component are left to the other domain-specific specifications and out of scope of this document. Domain-specific properties can be attached as child elements using [[XML-ns](#)].

## Value Component

Provides the value of the vouchers. There are two types of

values, i.e., fixed and ratio values. For a fixed value, the currency and amount of the value is specified. For a ratio value, the discount ratio of the price of the corresponding merchandise is specified.

Using the above Components, monetary meaning for diverse types of vouchers can be defined as shown in Table 1.

| Examples         | Number<br>needed for<br>redemption | Merchandise     | Value |        |          |
|------------------|------------------------------------|-----------------|-------|--------|----------|
|                  |                                    |                 | Ratio | Fixed  |          |
|                  |                                    |                 |       | Amount | Currency |
| Gift certificate | 1                                  | (Not specified) |       | 25     | USD      |
| Loyalty point    | 20                                 | (Not specified) |       | 200    | AUD      |
| Member card      | 0                                  | (Not specified) | 0.2   |        |          |
| Coupon           | 1                                  | Beef 500g       | 0.3   |        |          |
| Event ticket     | 1                                  | Hall A, S, K23  | 1.0   |        |          |
| Exchange ticket  | 1                                  | ISBN:0071355014 | 1.0   |        |          |

Table 1. Examples of vouchers and their properties

## 5. Syntax Overview and Examples

This section provides an overview and examples of Voucher Component. The formal syntax and semantics are found in Sections 6 and 7.

Voucher Components are represented by the <Voucher> element which has the following structure (where "?" denotes zero or one occurrence; "+" denotes one or more occurrences; and "\*" denotes zero or more occurrences):

```
<Voucher>
  (Provider)
  (Issuer)?
  <Promise>
    (Title)?
    (Description)?
    (ValidPeriod)?
    (Redemption)?
    (Value)?
    (Merchandise)+
  </Promise>
  (Holder)?
</Voucher>
```

An example of a Voucher Component is described below. This is an example of a five dollar discount coupon for specific merchandize, a book with ISBN number 0071355014. The coupon is valid from April

1st in 2001 to March 31st in 2002. To claim this offer, one voucher must be spent.

```
<?xml version="1.0"?>
<Voucher xmlns="http://www.ietf.org/rfc/rfcXXX.txt"
  xmlns:vts="http://www.example.com/vts.txt">
  <Provider Name="Voucher Wallet 2001">
    <vts:KeyInfo>...</vts:KeyInfo>
  </Provider>
  <Issuer Name="Alice Book Center, Ltd.">
    <vts:KeyInfo>...</vts:KeyInfo>
  </Issuer>
  <Promise>
    <Title>IOTP Book Coupon</Title>
    <Description>$5 off IOTP Book</Description>
    <ValidPeriod start="2001-04-01" end="2002-03-31"/>
    <Redemption spend="1"/>
    <Value currency="USD" amount="5"/>
    <Merchandise xmlns="http://www.example.com/book.txt">
      <BookID ISBN="0071355014"/>
    </Merchandise>
  </Promise>
</Voucher>
```

## **6. Semantics**

(tbs)

## **7. DTD**

(tbs)

## **8. Security Considerations**

Security issues for delivering Voucher Components are discussed in [Section 3](#). Security is a major issue in implementing VTS. For XML Voucher, however, the only requirements for achieving security are to provide the parameters needed for establishing security.

## **9. Acknowledgement**

(tbs)

## **10. References**

[ECML] ECML Version 2, to appear.

[GVT] K. Fujimura, "Requirements for Generic Voucher Trading",

[draft-ietf-trade-drt-requirements-02.txt](#), February 2001.

[IOTP] D. Burdett, "The Internet Open Trading Protocol", [RFC2801](#),

K. Fujimura, M. Terada

[Page 7]

April 2000.

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

[TLS] T. Dierks, C. Allen, "The TLS Protocol Version 1.0", [RFC2246](#), January 1999.

[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", [draft-ietf-xmlsig-core-11.txt](#), in RFC Editor queue for publication as Proposed Standard.

[XML-ns] "Namespaces in XML", A W3C Recommendation, <<http://www.w3.org/TR/REC-xml-names>>, January 1999.

## **11. Authors Address**

Ko Fujimura and Masayuki Terada  
NTT Corporation  
1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa, 239-0847 JAPAN  
Phone: +81-(0)468-59-3814  
Fax: +81-(0)468-59-2241  
Email: [fujimura@isl.ntt.co.jp](mailto:fujimura@isl.ntt.co.jp), [terada@isl.ntt.co.jp](mailto:terada@isl.ntt.co.jp)

