



Payment Systems & Data Security Specifications Release 2009B

Data Proxy Specification Version 1.0

Issue 2

Issue 2 includes a minor update to the 'About HTNG' section.

September 24, 2009

Payment Systems & Data Security Workgroup

About HTNG

Hotel Technology Next Generation ("HTNG") is a nonprofit organization with global scope, formed in 2002 to facilitate the development of next-generation, customer-centric technologies to better meet the needs of the global hotel community. HTNG's mission is to provide leadership that will facilitate the creation of one (or more) industry solution set(s) for the lodging industry that:

- Are modeled around the customer and allow for a rich definition and distribution of hotel products, beyond simply sleeping rooms;
- Comprise best-of-breed software components from existing vendors, and enable vendors to collaboratively produce world-class software products encompassing all major areas of technology spending: hotel operations, telecommunications, in-room entertainment, customer information systems, and electronic distribution;
- Properly exploit and leverage a base system architecture that provides integration and interoperability through messaging; and that provides security, redundancy, and high availability;
- Target the needs of hotel companies up to several hundred properties, that are too small to solve the issues themselves;
- Will reduce technology management cost and complexity while improving reliability and scalability; and
- Can be deployed globally, managed remotely, and outsourced to service providers where needed.

In June 2005, HTNG announced the first-ever "Branding and Certification Program" for hotel technology. This program will enable vendors to certify their products against open HTNG specifications, and to use the "HTNG Certified" logo in their advertising and collateral materials.

It will enable hotels to determine which vendors have completed certification of their products against which specific capabilities, and the environments in which performance is certified. HTNG's vision is to achieve a flexible technical environment that will allow multiple vendors' systems to interoperate and that will facilitate vendor alliances and the consolidation of applications, in order to provide hotels with easily managed, continually evolving, cost-effective solutions to meet their complete technology needs on a global basis.

Copyright 2009, Hotel Technology Next Generation

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

For any software code contained within this specification, permission is hereby granted, free-of-charge, to any person obtaining a copy of this specification (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the above copyright notice and this permission notice being included in all copies or substantial portions of the Software.

Paragraph added on 27 May 2010:

Manufacturers and software providers shall not claim compliance with portions of the requirements of any HTNG specification or standard, and shall not use the HTNG name or the name of the specification or standard in any statements about their respective product(s) unless the product(s) is (are) certified as compliant to the specification or standard.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is granted for implementers to use the names, labels, etc. contained within the specification. The intent of publication of the specification is to encourage implementations of the specification.

This specification has not been verified for avoidance of possible third-party proprietary rights. In implementing this specification, usual procedures to ensure the respect of possible third-party intellectual property rights should be followed.

The names Hotel Technology Next Generation and HTNG, and logos depicting these names, are trademarks of Hotel Technology Next Generation. Permission is granted for implementers to use the aforementioned names in technical documentation for the purpose of acknowledging the copyright and including the notice required above. All other use of the aforementioned names and logos requires the permission of Hotel Technology Next Generation, either in written form or as explicitly permitted for the organizations members through the current terms and conditions of membership.

Table of Contents

1	DOCUMENT HISTORY	4
1.1	DOCUMENT CHANGES	4
2	DOCUMENT INFORMATION	5
2.1	DOCUMENT PURPOSE	5
2.2	SCOPE	5
2.3	AUDIENCE	5
2.4	OVERVIEW	5
2.5	INDUSTRY/DOCUMENT TERMS & ACRONYMS	5
2.6	REFERENCED DOCUMENTS	8
3	BUSINESS PROCESS	9
3.1	OVERVIEW	9
3.2	ROLES	9
3.2.1	Business Logic System	9
3.2.2	Electronic Payment Logic and Creation System (EPLACS)	9
3.2.3	Proxy Vault	10
3.2.4	Payment Processing System	10
3.3	BUSINESS-LEVEL OVERVIEW – USE CASES	10
3.3.1	Get a DataProxy	10
3.3.2	Get a Credit Card	10
4	COMMON CLASSES AND DATA ELEMENTS	11
4.1	PAYMENT CARD PROXY REQUEST	11
4.1.1	Usage Profile Table	11
4.1.2	Example Message	11
4.2	PAYMENT CARD PROXY RESPONSE	12
4.2.1	Usage Profile Table	12
4.2.2	Example Message	13
4.3	PAYMENT CARD REQUEST	13
4.3.1	Usage Profile Table	13
4.3.2	Example Message	13
4.4	PAYMENT CARD RESPONSE	14
4.4.1	Usage Profile Table	14
4.4.2	Example Message	15
5	MESSAGE IMPLEMENTATION	16
5.1	MESSAGE IMPLEMENTATION REQUIREMENTS	16
5.2	TECHNICAL USE CASES	16
5.2.1	New/Updated Customer Profile	16
5.2.2	New Reservation	17
5.2.3	DataProxy-enabled system communicates with non-DataProxy-enabled system	17
5.2.4	Customer Service at the merchant location needs to retrieve the full payment card information	18
5.3	BULK FILE PROCESSING	18
6	PAYMENT CARD INDUSTRY DATA SECURITY STANDARD (PCI-DSS)	19
6.1	PCI-DSS	19
6.2	PA-DSS	19
6.3	RELATIONSHIP BETWEEN PCI DSS AND PA-DSS	19
6.4	TO WHICH APPLICATIONS DOES PA-DSS APPLY?	19
6.5	COMMUNICATIONS AND SECURITY	20

1 Document History

1.1 Document Changes			
Version	Date	Author	Comments
0.1	26 Aug 2008	J. Newby	First Draft
0.2	10 Sept	J. Newby	Second Draft with initial group comments
0.3	18 Sept	S. Zloth	Third Draft
0.4	21 October	S. Zloth	Fourth Draft
0.5	13 November	A. Lubitz P. Whittle	Add 3.3.1.3 DCC Use Case at Check-in Add 3.3.1.6 DCC Use Case at Check-out Add 7.3.2.2 Technical Use Case Guest Check-in Add 6.3 – Data Proxy Port Implementations
0.6	17 November	S. Zloth	Incorporated Common Classes, removed sections previously identified as ok to remove
0.7	March 12, 2009	O. Thompson, S. Zloth	Removed obsolete sections, cleaned up doc, preliminary updates to Business Use Cases, updated PCI and Security section and DataProxy section from 3-6-09 FTF meeting
0.8	April 1, 2009	S. Zloth	Separated Payments into a different document. This document will be only for DataProxy functionality
0.9	April 22, 2009	S. Zloth	More cleanup, incorporated section 2 from Bob Lowe.
1.0	June 5, 2009	J. Rosamilia	Added sample messages and field definitions.
1.1	June 19, 2009	S. Zloth	More cleanup, added Business Use Cases, removed Test Cases
1.2	June 19, 2009	J. Rosamilia	Removed references to TerminalID. Added EncryptedCardData element.
1.3	June 24, 2009	A. Lubitz	Minor cleanup to Section 3.2 – Roles Updated Section 6.1 – Message Implementation Requirements
1.31	June 30, 2009	A. Lubitz	Updated Section 6.1 - Message Implementation Requirements, Added Response Messages
1.4	July 7, 2009	S. Zloth	Miscellaneous cleanup, removal of unused sections
1.5	July 7, 2009	J. Rosamilia	Added XML field definitions
1.6	July 14, 2009	S. Zloth	Added Terms and Definitions, Technical Use Cases and Bulk File processing
Issue 1	Sept 24, 2009		Initial Release
Issue 2	May 27, 2010		Minor update to 'About HTNG' section

2 Document Information

2.1 Document Purpose

The purpose of this document is to provide a specification for implementation of the HTNG open-standards solution for card based transactions (i.e. payments or authorizations) by Credit/Debit or other card types. This specification was developed by the Payment Systems & Data Security Workgroup to define how card data should flow securely between various systems used around the world that are dependent on or handle card data within a property (e.g. a Property Management System and a Payment Gateway) and work in harmony with other specifications also developed by HTNG groups. Care has been taken to ensure this specification caters for global payment requirements and intended for use in any region, e.g. the U.S. AsiaPac, EMEA etc.

2.2 Scope

The scope of this document includes, directly or by reference, all information required to implement the interface, described above. It does not include information needed to implement other specifications developed by other 3rd parties.

2.3 Audience

The primary intended audience of this document is a developer or system designer seeking to implement the interface specifications within their products. As this document also provides Business Level Use Cases the secondary audience is general business readers wishing to familiarize themselves with the interactions between POS and Gateways, especially in to understand data security concerns.

2.4 Overview

There are many different types of card that may be used or presented within the Hospitality sector ranging from Payment Cards (such as Credit, Debit and Charge cards) to Gift and Loyalty Cards. There are also many different types of transaction that can be performed by these cards and each could have their own different data requirements. This document will concern itself primarily with the needs of payments.

The various regions of the world could have different payment requirements and processes, the most common of which will be addressed.

Globally, data security is an issue so this document will adopt and respect the security measures being imposed within the card payments industry.

2.5 Industry/Document Terms & Acronyms

For the purpose of this document the following terms have been defined as follows:

Term	Definition
Acquirer	A principal member of Visa and MasterCard associations that acquires data relating to Merchant transactions for processing.
ATS	
Authorization	See Transaction Authorization
Authorization Reversal	A process used to effectively cancel a previously approved authorization. Processors and card issuers handle this process differently and thus have different requirements.
AVS	Address Verification System – process by which Merchants supply cardholder address information for non-swiped transactions. AVS is positioned to be used as a fraud prevention tool by ensuring the cardholder's billing address correctly matches what is provided at the POS.
BIN	The Bank Identification Number ranges form the first 10 digits of the PAN and allow not only the Issuer but the exact type of card to be identified as well (i.e. XYZ Bank Gold MasterCard). BIN ranges are very important for DCC as once the Issuer and Type of card are known, the currency of that card can also be determined.
Cardholder Data	Full magnetic stripe or ICC 'chip' data defined as the primary account number plus – cardholder name – or – expiration date – or – service code.
Chargeback	A credit card transaction that is in dispute either by the cardholder or the cardholder's bank. Merchants must be present chargeback defense in order to validate the original charge. This may include information such as an invoice, receipt, restaurant check and customer signature.
Credit Card Security Code	Generic term used to identify the data elements that are used to protect a credit card against counterfeiting and tampering. There are two types of Credit Card Security Codes – one which is securely encrypted, stored on the magnetic stripe and is a protective element in card present transactions. The other code is a three or four digit number presented unembossed and it provides a layer of protection in card <u>not</u> present transactions. With

	<p>the exception of American Express the visible Credit Card Security Code is found near the card's signature panel while on an American Express card it is on the face of the card.</p> <p>See below for specific card brand code names.</p> <p>Magnetic Stripe Codes</p> <p>CVV – Card Verification Value (Visa and Discover) CVC – Card Validation Code (MasterCard) CAV - Card Authentication Value (JCB) CSC – Card Security Code (American Express)</p> <p>Visible Codes</p> <p>CID – Card Identification Number (American Express & Discover) CVV2 – Card Authentication Value 2 (Visa) CVC2 – Card Validation Code 2 (MasterCard) CAV2 – Card Authentication Value 2 (JCB)</p>
CRM	Central Reservation System -
Data Encryption	Process of converting information into an unintelligible form except to holders of a specific cryptographic key. Use of encryption protects information between the encryption process and the decryption process (the inverse of encryption) against unauthorized disclosure
Data Proxy	A sequence of characters that acts as a data reference to a primary account number (credit card number). The use of a data proxy eliminates the need to store credit card information across many disparate systems.
DCC	D ynamic C urrency C onversion – is a service which allows hoteliers or merchants to offer to consumers the option to complete their purchase in cardholder (issuing bank) currency. This is offered as a customer service to international guests by providing currency pricing transparency at the point-of-sale. The merchant prices their goods or services in local currency. Following the card swipe a currency eligibility determination is made whereupon the consumer is offered the option of completing their transaction in their home currency based upon the conversion performed at time-of-sale. Visa and MasterCard have specific rules with respect to offering of DCC – specifically regarding customer opt-in and the inclusion of certain information and disclosure language used on their respective receipts.
Decline	Where the card Issuer or the Processor can not offer Authorization to a transaction request for what ever reason (i.e. insufficient funds, stolen card, invalid card etc.)
EMEA	E urope, M iddle E ast and A frica – generic trading zone.
EMV	The acronym EMV stands for E uropay, M asterCard and V isa. The three aforementioned companies created the standard known widely as Chip & PIN card processing. Chip & PIN or EMV is a growing, globalized methodology to securing credit card data.
EPLACS	E lectronic P ayment L ogic A nd C reation S ystem – a system that applies the card association rules to business transactions to ensure conformance to qualify for optimal interchange.
Gateway	A system used to transmit transaction data to and from the Processor/Acquirer.
Guest	The cardholder requesting goods or services from the property.
ICC	I ntegrated C hip C ard – plastic card with and embedded micro chip that contained the unique card data (see EMV).
IIN	The I ssuer I dentification N umber makes up the first 6 digits of the PAN to help identify the type of card scheme and the Issuer (i.e. XYZ Bank MasterCard). The IIN is often confused or referred to as the BIN.
Incremental Authorization	A process used to add additional authorization to a card as additional charges incurred during the course of a guest's stay pushes the total charge amount above the previously attained authorization amount.
Interchange	The exchange of information, transactional data and money among banks. Interchange systems are managed by Visa and MasterCard associations and are standardized so that banks and merchants across the globe can use them.
Interchange fees	This is the fee that the Card Association charges the Merchant to get the funds into his bank (Merchant Bank) and to get the billing information to the Cardholder's Bank (Issuing Bank). Interchange Fees are based on following credit card regulations and capturing appropriate data including card swipe, address, and electronic signature as needed. These fees are also based on the timeliness of the settlement of transactions.
Issue Number	The Issue Number is a component of a UK Issued Debit card (Maestro) where the card holder's actual bank account number is embedded as part of the PAN. If the card is replaced the Issue Number (up to 2 digits) is incremented and the PAN remains the same. The Issue Number can be read from the Track2 and is also embossed on the front of the card.
Issuer	The financial institution, usually banks, that issue credit cards to individuals with

	<p>agreements for repayment. These financial institutions promote the use of the various branded cards and charge the cardholders interest and fees for their use. They share in the Interchange Fee charged by the Card Associations. Most of the power in the credit card industry is seated with the Issuing Banks. An Issuing Banks' worth is its portfolio of cardholders.</p>
Kiosk	An unattended point of sale (such as a Vending Machine) where the cardholder may perform a transaction.
Luhn Check Digit	The Luhn Check Digit is the last digit of the PAN (which is calculated using Modulus 10) to help safeguard against incorrect manual entry of the card number.
Magnetic Strip (Track) Data	<p>Data encoded in the magnetic stripes and used for authorization during card present transactions. PCI guidelines state that systems cannot retain full magnetic stripe data subsequent to transaction authorization. Data such as account number, expiration date, name, and service code may be extracted and retained, if needed for business purposes.</p> <p>There are three magnetic stripe tracks on credit cards.</p> <p>Track 1 can hold up to 79 characters, six of which are reserved control characters. Data encoded on track 1 includes - PAN, country code, full name, expiration date, and potentially other discretionary data.</p> <p>Track 2 can hold up to 40 characters. Typical track 2 data includes PAN, expiration date, and discretionary data.</p> <p>Track 3 allows up to 107 characters to be encoded and is widely recognized as a "read/write" track while track's 1 & 2 are read only. Track 3 was originally designed to support ATM functions but is rarely used any longer for electronic transactions.</p>
Merchant MID	<p>The organization or business providing goods or services to the cardholder.</p> <p>Merchant Identifier - the bank/acquirer allocated value to refer to the organization or business given the agreement to accept card transactions.</p>
Multi-currency	<p>Multi-currency Pricing is a service / technique which allows hoteliers or merchants to price their goods or services in currencies other than the local currency. MCP is not the same as a 'pricing calculator' which provides an indicative home currency amount to consumers. Using MCP, a merchant is able to provide pricing in foreign currencies and accept payment from the consumer in the currency selected by the consumer. MCP is distinguished from DCC in that the pricing is offered in foreign currencies in advance of the consumer swiping or entering their card number. There are no specific Visa or MasterCard rules relating to MCP beyond the standard receipt formatting and messaging requirements.</p>
NFC	Near Field Communications – see RFID
Open to Buy	Industry term referring to the difference between the cardholder's credit limit and the sum of holds (authorizations) and outstanding balance. Lodging holds (authorizations) can dramatically affect a card-holders "open to buy" amount. Some card associations are mandating the use of authorization reversals to better manage this issue.
PA-DSS	Payment Application Data Security Standard formerly referred to as PABP.
PAN	Primary Account Number also known as the payment card number that identifies the issuer and the primary account cardholder.
Payment Application	Software programs that either store, process or transmit PAN as part of the authorization or settlement process.
PCI DSS	Payment Card Industry Data Security Standard
PCI SSC	Payment Card Industry Security Standard's Council – the organization that was founded by the major card issuers whose mission was to improve cardholder data security. The primary responsibility of this organization is to maintain the PCI, PED and Payment Application Data Security Standards.
PED	PIN Entry Device – a common name for a chip card reader that has a built in PIN pad.
PIN	Personal Identification Number – primarily used with debit and/or Chip card transactions
PMS	Property Management System – a system used in the Lodging Industry to manage all aspects of a hotel and potentially used to accept the cardholder data at check-in/check-out.
POS	Pont Of Sale – the terminal or cash register where the transaction is being performed.
QSA	Qualified Security Assessor – PCI SSC trained and approved entities capable of performing PCI and PA-DSS audits as well as other security functions.
Referral	Where the Processor or Acquirer requires more information to complete an Authorization Request (i.e. is it the genuine card holder performing the transaction). This typically involves a voice call to the merchant or the cardholder at the point of sale.
Refund/Credit	The process of returning credit to a cardholder account for reasons such as an erroneous charge or a returned item.
RFID	Radio Frequency Identification – an alternative medium to uniquely identify a cardholder as opposed to a magnetic stripe or ICC Data. Requires the device (card) to be merely in near proximity of the reader rather than in direct contact with it.
Sensitive Data	e.g. the Track2, Expiry Date, AVS information open to compromise.

Service Establishment	See Merchant (American Express terminology)
Settlement	Credit card settlement is the process by which previously authorized transactions are submitted to card issuers for the merchant to receive payment.
Start Date	The date from which the card is valid. Displayed in MM/YY format and may be embossed on the front of the card.
Swipe	The action of passing the card through a Magnetic Card Reader to electronically capture the card data from the magnetic stripe on the back of the card.
TID	Terminal Identifier – the bank/acquirer allocated value to refer to an actual point of sale.
Track 2	See Magnetic Stripe
Transaction Amount	The monetary value in the local currency for the transaction being performed by the cardholder.
Transaction Authorization	Authorization is the process by which card issuer either approves, refers or denies requests to accept transactions. Approval is based on a validation of the account number and expiration date to verify that a cardholder's account is open, and that the transaction will not place the account above any credit limit. Since most authorization requests are approved, the term "authorized transaction" refers to an approved authorization request.
Transaction Void	A voided transaction describes a situation where a credit card transaction has been effectively deleted prior to settlement and/or close
Vault	The secure environment/system being used to store and protect cardholder data.
Web Services	A method of accepting and validating data from a remote system, typically over the Internet.

2.6 Referenced Documents

The following table shows the documents upon which this document depends:

Name	Location
PCI-DSS documentation	
ISO currency 4271	
Visa Operating Regulations	

3 Business Process

3.1 Overview

The specification is organized into two parts, Payments Processing Specification and Data Proxy Specification. The Payments Processing Specification describes the messages that enable a Point Of Sales System, POS, to process Lodging Industry Payment Card transactions with Payment Gateways or Acquiring Banks. The Data Proxy Specification defines the messages that allow the storage of sensitive cardholder information to be moved from the POS system to a secure data vault. The POS system exchanges the actual payment card detail for a Data Proxy or Token at the time the card information would have entered the POS. The POS then uses the Data Proxy within messages defined in the Payments Processing Specification to perform Payment Transactions.

Organizations may choose to provide a Payment Processing Service, or a Data Proxy Service or both.

3.2 Roles

In HTNG's web services implementation, the different "end points" of each interface are defined by roles. Each product implements at least one role, and in a complex specification that defines many roles, may implement multiple roles. HTNG certification to this specification requires the implementation of at least one role.

The use of roles allows any existing system, such as a PMS, CRS, Profile Management System, etc., to implement whatever functionality it deems appropriate, while still conforming to this specification. Each such system need only evaluate which roles its functionality plays in relevant transactions, and needs only to implement the required interface transactions that are defined for those roles. Hence, for example, the specification does not set forth specific requirements for a Property Management System, but rather for the payment data roles performed by that system. A Property Management System that performs more business functions than another may, as a consequence, need to handle more payment data roles.

Any commercial system may play multiple roles, and if it does so, it must implement the required functionality of each role with respect to external messaging interfaces, but it may handle internal transactions among the roles in any manner of its choosing.

This specification defines the following roles.

3.2.1 Business Logic System

A Business Logic System is a system used by the merchant to manage key elements of its business, such as reservations, check-in, profile maintenance, etc., and that requires the handling and/or transmission of sensitive payment data. If storage of sensitive payment data is required, it must use a DataProxy as a replacement for the sensitive data.

Examples of Business Logic Systems include the following. It should be noted that examples represent common implementations commonly found in the hospitality industry and are provided for ease of understanding, but by no means are meant to be all-inclusive.

- Central Reservation Systems
- Customer Relationship Management Systems
- Profile Management Systems
- Property Management Systems
- Global Distribution Systems
- Distribution Gateways and Switches
- Kiosks (depending on design)

3.2.2 Electronic Payment Logic and Creation System (EPLACS)

An Electronic Payment Logic and Creation System applies the card association rules to business transactions to ensure conformance. It may also apply logic to attempt to qualify for optimal interchange rates (for example, determining when to apply for an incremental authorization or a reversal) and to conform to the data requirements of particular card schemes. It consolidates data needed to process an electronic payment, including a transaction amount obtained from a Business Logic System, and prepares that data for submission to a Payment Processing System for authorization or settlement. An EPLACS typically needs to retain transaction data for at least a limited period of time, such as the duration of a guest stay. Where sensitive transaction data is retained, use of a DataProxy is required.

In typical commercial installations, the EPLACS may be a module within a property management system, a payment gateway, or a combination.

3.2.3 Proxy Vault

A secure system that stores sensitive data in a secure manner, preferably in an encrypted form. It manages the relationship between sensitive data and a proxy that represents that data, and provides methods for retrieval of sensitive data using the proxy as a retrieval key.

In typical commercial implementations of a Proxy Vault, it may be a component of the property management system or a gateway.

3.2.4 Payment Processing System

A system or systems that provide authorization and clearing of electronic payments.

Typically this role is delivered by a combination of systems that commonly include the following: gateway, merchant processor, card association, merchant acquirer, issuing bank. These systems, unlike the systems that provide the other roles described herein, are typically not under the control of merchants.

In typical commercial implementations, the Payment Processing System will receive messages that contain sensitive data. These messages may be sent by the Electronic Payment Logic and Creation System, the Gateway or a combination of both. In some areas of the world (e.g. Malaysia), the full message sent by the Electronic Payment Logic and Creation System, or the Gateway may be encrypted.

3.3 Business-Level Overview – Use Cases

3.3.1 Get a DataProxy

For any interaction between a system that has the credit card and needs to exchange it for a DataProxy (i.e., guest enters/updates their profile and supplies a new credit card, guest makes a reservation with a credit card, etc.):

- Originating system sends request to DataProxyVault requesting DataProxy
- Vault replies with DataProxy
- Originating system stores DataProxy in place of credit card

3.3.2 Get a Credit Card

For any interaction between a system that has the DataProxy and needs to exchange it for a Credit card number (i.e. Guest checks in at a DataProxy-enabled PMS, but needs to send reservation information to a non-DataProxy-enabled third party system):

- Originating system sends request to DataProxyVault requesting credit card information
- Vault replies with credit card information

4 Common Classes and Data Elements

4.1 Payment Card Proxy Request

To obtain a proxy, the HTNG_PaymentCardProxyRQ message is sent.

4.1.1 Usage Profile Table

Element @Attribute	Num	Description/Contents
HTNG_PaymentCardProxyRQ	1	Root element of the message.
@EchoToken	0..1	This is a value that is randomly generated by the sending system. The receiving system should return this value in its response so the original sender can match the original request with the response. This is especially important when communications is asynchronous.
@TimeStamp	1	Time of the transaction.
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message.
HTNG_PaymentCardProxyRQ / POS	1	Must be sent for the message to have meaning.
HTNG_PaymentCardProxyRQ / POS / Source	1	Must be sent for the message to have meaning.
HTNG_PaymentCardProxyRQ / POS / Source / RequestorID	1	Must be sent for the message to have meaning.
@Type	0..1	Refers to OTA code list UIT (Unique Id Type)
@ID_Context	1	The origin or an agreed-upon context in which the ID was either issued or otherwise further qualifies the ID.
@ID	1	The identifier of the system transmitting the request.
HTNG_PaymentCardProxyRQ / POS / Source / RequestorID / CompanyName	0	Optional, but must be present if @CompanyShortName is to be used.
@CompanyShortName	1	A easily recognizable, human-readable string that gives meaning to the RequestorID@ID
HTNG_PaymentCardProxyRQ / PaymentCards	1	Must be sent for the message to have meaning.
HTNG_PaymentCardProxyRQ / PaymentCards / PaymentCard	1..n	Must be sent for the message to have meaning.
@CardNumber	1	The Primary Account Number to be secured by the Data Proxy Service.
@CardCode	0..1	The two letter code indicative of the card (VS, MC, AX, etc.)
@ExpireDate	1	The expiration date for the Primary Account Number.
HTNG_PaymentCardProxyRQ / PaymentCards / PaymentCard / EncryptedCardData	0..1	Magnetic data which was obtained from a device that performed secure, real-time encryption, of the track data.
HTNG_PaymentCardProxyRQ / PaymentCards / PaymentCard / CardHolderName	1	The name of the card holder, as it appears on the front of the card.

4.1.2 Example Message

Scenario:

Property Management System "ABC" needs to obtain a payment card proxy for a new card just entered into the PMS. It transmits the following data to the Data Proxy Provider:

```
<HTNG_PaymentCardProxyRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="1.000"
xsi:schemaLocation="http://htng.org/2009B HTNG_PaymentCardProxyRQ.xsd" xmlns="http://htng.org/2009B"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <POS>
    <Source>
      <RequestorID Type="0" ID_Context="a" ID="a">
        <CompanyName CompanyShortName="a"></CompanyName>
      </RequestorID>
    </Source>
  </POS>
</HTNG_PaymentCardProxyRQ>
```

```

</Source>
</POS>
<PaymentCards>
  <PaymentCard CardNumber="4444333322221111" CardCode="VI" ExpireDate="1214">
    <EncryptedCardData>ynoeytcepqnyt4pqyt39p28y835y8[mncyr89pb8[vabr6y3896btv6e76c7b7
    65v36o7bc36a7o9xn6qt7647toq46n0cxt64q98qaxo6yq76n32QY3MQ6Y379RMT964PRMTYT98A, YGLIHN
    EGYMLUGJJEZ;GIP;GUSZMOGCOUIYWNAOCTOY489QMTCEY79ATYC98AX, TYU8IYG8RO9PIGU, WS89YU9P, uoi
    utp98, uo8pm, ty4op8i, tciothyzouimycuyt8ptcm, iohriosp; r9tui94tu9; mt4sa9tucmklgjfd; l.g
    j09sehmsvp[98usysptmorjgfd; hjfdlkbnopimxcojooz. ;utt0[9rmvreishtuj; irsicyu589ysgiuy
    c9pt40pwr87y95yum5; po687wq9tmna[ t9urusvn; ovt5wm9u7apua94375t432u6mvytr98osctk5cukw8
    typ8turefgildljgriogufjkghjr; p[9t84590yu5phg8jhysoiherkoltygoroa jkto pap9[ut5wm[8vt[
    4wv, yu590wy[85y875mvy75w[cm490c6t75w90c[y5wy9u7c8ptym4uhtrnt5iuthruighr; ysh8tp4, uwc
    tj4qicortyeaop; tu43qioty3quoiry4fw</EncryptedCardData>
    <CardHolderName>John A Smith</CardHolderName>
  </PaymentCard>
</PaymentCards>
</HTNG_PaymentCardProxyRQ>

```

4.2 Payment Card Proxy Response

The HTNG_PaymentCardProxyRQ is answered using the HTNG_PaymentCardProxyRS message.

4.2.1 Usage Profile Table

Element @Attribute	Num	Description/Contents
HTNG_PaymentCardProxyRS	1	Root element of the message.
@EchoToken	0..1	This is a value that is randomly generated by the sending system. The receiving system should return this value in its response so the original sender can match the original request with the response. This is especially important when communications is asynchronous.
@TimeStamp	1	Time of the transaction.
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message.
HTNG_PaymentCardProxyRS / Success	0..1	This is the annotation that the PaymentCard was able to be lodged by the receiving system and that a ProxyID for each card was able to be issued. It could be combined with Warning elements to denote a non-critical failure.
HTNG_PaymentCardProxyRS / Warnings	0..1	A collection of Warning elements.
HTNG_PaymentCardProxyRS / Warnings / Warning	1	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type = "Unknown". Refer to OTA Code List Error Warning Type (EWT).
@Language	0..1	Identifies the language used in textual descriptions.
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed Incomplete Complete Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@RecordID	0..1	If the receiving system is able to identify within a batch of PaymentCards which specific PaymentCard failed, the PaymentCard@CardNumber should be reported here.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OTA Code List Error Codes (ERR).
HTNG_PaymentCardProxyRS / ProxyIDs	1	A collection of ProxyID elements.
HTNG_PaymentCardProxyRS / ProxyIDs / ProxyID	1..n	The proxy, or stand-in value, that represents a Primary Account Number.

4.2.2 Example Message

Scenario:

The Data Proxy Provider answers the request by providing the following response:

```
<HTNG_PaymentCardProxyRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="1.000"
xsi:schemaLocation="http://htng.org/2009B HTNG_PaymentCardProxyRS.xsd" xmlns="http://htng.org/2009B"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Success/>
  <Warnings>
    <Warning Type="0" Language="en-us" Status="a" RecordID="a" ShortText="a"
Code="0">String</Warning>
  </Warnings>
  <ProxyIDs>
    <ProxyID>7617653287451111</ProxyID>
  </ProxyIDs>
</HTNG_PaymentCardProxyRS>
```

4.3 Payment Card Request

To obtain the payment card information for a previously obtained payment card proxy, the HTNG_PaymentCardRQ message is used.

4.3.1 Usage Profile Table

Element @Attribute	Num	Description/Contents
HTNG_PaymentCardRQ	1	Root element of the message.
@EchoToken	0..1	This is a value that is randomly generated by the sending system. The receiving system should return this value in its response so the original sender can match the original request with the response. This is especially important when communications is asynchronous.
@TimeStamp	1	Time of the transaction.
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message.
HTNG_PaymentCardRQ / POS	1	Must be sent for the message to have meaning.
HTNG_PaymentCardRQ / POS / Source	1	Must be sent for the message to have meaning.
HTNG_PaymentCardRQ / POS / Source / RequestorID	1	Must be sent for the message to have meaning.
@Type	0..1	Refers to OTA code list UIT (Unique Id Type)
@ID_Context	1	The origin or an agreed-upon context in which the ID was either issued or otherwise further qualifies the ID.
@ID	1	The identifier of the system transmitting the request.
HTNG_PaymentCardProxyRQ / POS / Source / RequestorID / CompanyName	0	Optional, but must be present if @CompanyShortName is to be used.
@CompanyShortName	1	A easily recognizable, human-readable string that gives meaning to the RequestorID@ID
HTNG_PaymentCardRQ / ProxyIDs	1	A collection of ProxyID elements.
HTNG_PaymentCardRQ / ProxyIDs / ProxyID	1..n	The proxy, or stand-in value, that represents a Primary Account Number.

4.3.2 Example Message

Scenario:

Property Management System "ABC" obtains the card information for a card previously stored by the Data Proxy Provider by transmit the following message:

```
<HTNG_PaymentCardRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="1.000"
xsi:schemaLocation="http://htng.org/2009B HTNG_PaymentCardRQ.xsd" xmlns="http://htng.org/2009B"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <POS>
```

```

    <Source>
      <RequestorID Type="0" ID_Context="a" ID="a">
        <CompanyName CompanyShortName="a"></CompanyName>
      </RequestorID>
    </Source>
  </POS>
  <ProxyIDs>
    <ProxyID>7617653287451111</ProxyID>
  </ProxyIDs>
</HTNG_PaymentCardRQ>

```

4.4 Payment Card Response

The HTNG_PaymentCardRQ is answered using the HTNG_PaymentCardRS message.

4.4.1 Usage Profile Table

Element @Attribute	Num	Description/Contents
HTNG_PaymentCardRS	1	Root element of the message.
@EchoToken	0..1	This is a value that is randomly generated by the sending system. The receiving system should return this value in its response so the original sender can match the original request with the response. This is especially important when communications is asynchronous.
@TimeStamp	1	Time of the transaction.
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
HTNG_PaymentCardRS / Success	0..1	This is the annotation that the PaymentCard was able to be retrieved via the ProxyID from the request. It could be combined with Warning elements to denote a non-critical failure.
HTNG_PaymentCardRS / Warnings	0..1	A collection of Warning elements.
HTNG_PaymentCardRS / Warnings / Warning	1	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type = "Unknown". Refer to OTA Code List Error Warning Type (EWT).
@Language	1	Identifies the language used in textual descriptions.
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed Incomplete Complete Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@RecordID	0..1	If the receiving system is able to identify within a batch of ProxyIDs which specific ProxyID failed, the ProxyID should be reported here.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OTA Code List Error Codes (ERR).
HTNG_PaymentCardRS / PaymentsCards	1	Must be sent for the message to have meaning.
HTNG_PaymentCardRS / PaymentCards / PaymentCard	1..n	Must be sent for the message to have meaning.
@CardNumber	1	The Primary Account Number representing the ProxyID.
@CardCode	1	The two letter code indicative of the card (VS, MC, AX, etc.)
@ExpireDate	1	The expiration date for the Primary Account Number.
HTNG_PaymentCardRS / PaymentCards / PaymentCard / CardHolderName	0..1	The name of the card holder, as it appears on the front of the card.

4.4.2 Example Message

Scenario:

The Data Proxy Provider answers the request by providing the following response:

```
<HTNG_PaymentCardRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="1.000"
xsi:schemaLocation="http://htng.org/2009B HTNG_PaymentCardRS.xsd" xmlns="http://htng.org/2009B"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Success/>
  <Warnings>
    <Warning Type="0" Language="en-us" Status="a" RecordID="a" ShortText="a"
Code="0">String</Warning>
  </Warnings>
  <PaymentCards>
    <PaymentCard CardNumber="4444333322221111" CardCode="VI" ExpireDate="1214">
      <CardHolderName>John A Smith</CardHolderName>
    </PaymentCard>
  </PaymentCards>
</HTNG_PaymentCardRS>
```

5 Message Implementation

5.1 Message Implementation Requirements

Each role requires the implementation of certain messages, and some roles permit the optional support of additional messages. To certify a product for any particular role, all messages specified as mandatory for that role must be supported.

When an optional message is not supported by a system, but where a return value is to be provided, the system should return the result of "SUCCESS" so that the sending system does not perceive the transaction to have caused an error.

The following chart represents **Web Services Operations that will be provided (supported) by these systems**

Message	WSDL Operation	Business Logic System	EPLACS	Proxy Vault	Payment System	Processing
Request DataProxy	HTNG_PaymentCardProxyRQ	Opt	Opt	√		
Response DataProxy	HTNG_PaymentCardProxyRS	Opt	Opt	√		
Request Card Data	HTNG_PaymentCardRQ	Opt	Opt	√		
Response Card Data	HTNG_PaymentCardRS	Opt	Opt	√		

Opt denotes *Optional Web Services*

√ denotes Required

The following chart represents Web Service Operations which systems will **initiate these messages**.

Message	WSDL Operation	Business Logic System	EPLACS	Proxy Vault	Payment System	Processing
Request DataProxy	HTNG_PaymentCardProxyRQ	Opt	Opt			
Response DataProxy	HTNG_PaymentCardProxyRS	Opt	Opt	√		
Request Card Data	HTNG_PaymentCardRQ	Opt	Opt			
Response Card Data	HTNG_PaymentCardRS	Opt	Opt	√		

Opt denotes *Optional Web Services*

√ denotes Required

5.2 Technical Use Cases

5.2.1 New/Updated Customer Profile

ID	DP01
Provider	Business Logic System
Actor	Payment Logic of Business Logic System

5.2.1.1 Brief Description

When a guest enters a payment card in a Customer Profile system, that card needs to be exchanged for a DataProxy.

5.2.1.2 Basic Flow

1. The use case starts when the actor identifies a guest is entering a payment card.
2. The Actor issues a **HTNG_PaymentCardProxyRQ** to the Proxy Vault and will receive a **HTNG_PaymentCardProxyRS** in return.
3. The use case terminates.

5.2.1.3 Preconditions

1. None

5.2.1.4 Postconditions

2. The DataProxy is stored with the guest's profile record as a replacement for the payment card number.

5.2.2 *New Reservation*

ID	DP02
Provider	Business Logic System
Actor	Payment Logic of Business Logic System

5.2.2.1 Brief Description

Guest makes a new reservation using a payment card that has not been exchanged for a DataProxy.

5.2.2.2 Basic Flow

1. The use case starts when the actor identifies a guest is entering a payment card.
2. The Actor issues a **HTNG_PaymentCardProxyRQ** to the Proxy Vault and will receive a **HTNG_PaymentCardProxyRS** in return.
3. The use case terminates.

5.2.2.3 Preconditions

1. None

5.2.2.4 Postconditions

1. The DataProxy is stored with the guest's reservation record as a replacement for the payment card number.

5.2.3 *DataProxy-enabled system communicates with non-DataProxy-enabled system*

ID	DP03
Provider	Business Logic System
Actor	Payment Logic of Business Logic System

5.2.3.1 Brief Description

If a system that is DataProxy-enabled needs to send payment card information to a system that is non-DataProxy-enabled, it needs to retrieve the payment card information from the Proxy Vault.

5.2.3.2 Basic Flow

1. The use case starts when the actor identifies a request for payment card information.
2. The Actor issues a **HTNG_PaymentCardRQ** to the Proxy Vault and will receive a **HTNG_PaymentCardRS** in return.
3. The use case terminates.

5.2.3.3 Preconditions

1. None

5.2.3.4 Postconditions

1. None.

5.2.4 Customer Service at the merchant location needs to retrieve the full payment card information

ID	DP04
Provider	Business Logic System
Actor	Payment Logic of Business Logic System

5.2.4.1 Brief Description

Merchant's Customer Service needs to retrieve the full card number in order to respond to a chargeback or inquiry.

5.2.4.2 Basic Flow

1. The use case starts when the actor identifies a request for payment card information.
2. The Actor issues a **HTNG_PaymentCardRQ** to the Proxy Vault and will receive a **HTNG_PaymentCardRS** in return.
3. The use case terminates.

5.2.4.3 Preconditions

1. None

5.2.4.4 Postconditions

1. None.

5.3 Bulk File Processing

This workgroup understands that there may be use cases that require the processing of large volume or bulk set of credit cards numbers. While the methods we have defined in this document would allow the processing of these credit numbers one at a time, they would prove to be inefficient in these large volumes.

Some examples are:

- A merchant who has only DataProxies needs to transmit folio information to a 3rd party provider who requires credit card data
- A merchant needs to retrieve all credit card numbers it has on file with the Vault provider
- A merchant requiring the transfer of credit card numbers from proxies system to non-proxy system.

The workgroup will address this need in the next release of this specification.

6 Payment Card Industry Data Security Standard (PCI-DSS)

6.1 PCI-DSS

The PCI DSS, a set of comprehensive requirements for enhancing payment account data security, was developed by the founding payment brands of the PCI Security Standards Council, including American Express, Discover Financial Services, JCB International, MasterCard Worldwide and Visa Inc. Inc. International, to help facilitate the broad adoption of consistent data security measures on a global basis.

The PCI DSS is a multifaceted security standard that includes requirements for security management, policies, procedures, network architecture, software design and other critical protective measures. This comprehensive standard is intended to help organizations proactively protect customer account data.

The PCI Security Standards Council will enhance the PCI DSS as needed to ensure that the standard includes any new or modified requirements necessary to mitigate emerging payment security risks, while continuing to foster wide-scale adoption.

Integrators should use standard PCI-approved methods for communicating over private and public networks. Detailed information on what is required for PCI-DSS compliance can be found using the following link to the PCI Council web site:

<https://www.pcisecuritystandards.org/>

6.2 PA-DSS

PA-DSS is the Council-managed program formerly under the supervision of the Visa Inc. program known as the Payment Application Best Practices (PABP). The goal of PA-DSS is to help software vendors and others develop secure payment applications that do not store prohibited data, such as full magnetic stripe, CVV2 or PIN data, and ensure their payment applications support compliance with the PCI DSS. Payment applications that are sold, distributed or licensed to third parties are subject to the PA-DSS requirements.

Information on PA-DSS compliance can be found here:

https://www.pcisecuritystandards.org/security_standards/pa_dss.shtml

6.3 Relationship between PCI DSS and PA-DSS

The requirements for the Payment Application Data Security Standard (PA-DSS) are derived from the Payment Card Industry Data Security Standard (PCI DSS) Requirements and Security Assessment Procedures. This document, which can be found at www.pcisecuritystandards.org, details what is required to be PCI DSS compliant (and therefore what a payment application must support to facilitate a customer's PCI DSS compliance).

Traditional PCI Data Security Standard compliance may not apply directly to payment application vendors since most vendors do not store, process, or transmit cardholder data. However, since these payment applications are used by customers to store, process, and transmit cardholder data, and customers are required to be PCI Data Security Standard compliant, payment applications should facilitate, and not prevent, the customers' PCI Data Security Standard compliance. Just a few of the ways payment applications can prevent compliance follow.

1. Storage of magnetic stripe data in the customer's network after authorization;
2. Applications that require customers to disable other features required by the PCI Data Security Standard, like anti-virus software or firewalls, in order to get the payment application to work properly; and
3. Vendor's use of unsecured methods to connect to the application to provide support to the customer.

Secure payment applications, when implemented in a PCI DSS-compliant environment, will minimize the potential for security breaches leading to compromises of full magnetic stripe data, card validation codes and values (CAV2, CID, CVC2, CVV2), PINs and PIN blocks, and the damaging fraud resulting from these breaches.

6.4 To Which Applications does PA-DSS Apply?

For purposes of PA-DSS, a payment application is defined as one that stores, processes, or transmits cardholder data as part of authorization or settlement, where the payment applications is sold, distributed, or licensed to third parties.

The following guide can be used to determine whether PA-DSS applies to a given payment application:

PA-DSS does apply to payment applications that are typically sold and installed "off the shelf" without much customization by software vendors.

PA-DSS does apply to payment applications provided in modules, which typically includes a "baseline" module and other modules specific to customer types or functions, or customized per customer request. PA-DSS may only apply to the baseline module if that module is the only one performing payment functions (once confirmed by a PA-QSA). If other modules also perform payment functions, PA-DSS applies to those modules as well. Note that it is considered a "best practice" for software vendors to isolate payment functions into a single or small number of baseline modules, reserving other modules for non-payment functions. This best practice (though not a requirement) can limit the number of modules subject to PA-DSS.

PA-DSS does NOT apply to a payment application developed for and sold to only one customer since this application will be covered as part of the customer's normal PCI DSS compliance review.

Note that such an application (which may be referred to as a "bespoke" application) is sold to only one customer (usually a large merchant or service provider), and it is designed and developed according to customer-provided specifications.

PA-DSS does NOT apply to payment applications developed by merchants and service providers if used only in-house (not sold, distributed, or licensed to a third party), since this in-house developed payment application would be covered as part of the merchant's or service provider's normal PCI DSS compliance.

For example, for the last two bullets above, whether the in-house developed or "bespoke" payment application stores prohibited sensitive authentication data or allows complex passwords would be covered as part of the merchant's or service provider's normal PCI DSS compliance efforts and would not require a separate PA-DSS assessment.

The following list, while not all-inclusive, illustrates applications that are NOT payment applications for purposes of PA-DSS (and therefore do not need to undergo PA-DSS reviews):

- Operating systems onto which a payment application is installed (for example, Windows, Unix)
- Database systems that store cardholder data (for example, Oracle)
- Back-office systems that store cardholder data (for example, for reporting or customer service purposes)

6.5 Communications and Security

Integrators should use standard PCI-approved methods for communicating over private and public networks. In addition, because the DataProxyVault is storing a large volume of sensitive data, we recommend that integrators use extra layer(s) of security for authenticating clients who are performing transactions that retrieve sensitive data from the DataProxy Vault.

These additional methodologies may include one or more of the following:

- Client-side digital certificates
- Merchant/Terminal ID application-layer filtering