



**Payment Systems & Data Security
Specifications, Release 2010A**

Payment Processing Specification Version 1.0

Issue 1

23 April 2010

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 2010, 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.

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	5
1.1	DOCUMENT CHANGES	5
2	DOCUMENT INFORMATION	6
2.1	DOCUMENT PURPOSE.....	6
2.2	SCOPE	6
2.3	AUDIENCE.....	6
2.4	OVERVIEW	6
2.5	INDUSTRY/DOCUMENT TERMS & ACRONYMS	6
2.6	REFERENCED DOCUMENTS.....	9
3	BUSINESS PROCESS.....	10
3.1	OVERVIEW	10
3.2	ROLES	10
3.2.1	Business Logic System	10
3.2.2	Payment Processing System.....	10
3.3	BUSINESS PROCESS FLOW	10
3.3.1	Reservation Process	11
3.3.2	Change Card Process.....	13
3.3.3	Guest Check-in Process	14
3.3.4	During Stay Activity	15
3.3.5	Guest Check-out Process	16
3.3.6	Extended Stay Processing	19
3.3.7	Cancellation/No-show.....	20
3.3.8	Batch Close.....	21
3.3.9	Void Settlement, marking a transaction as not "settleable"	21
3.3.10	Return Transaction.....	22
3.3.11	Post Departure Charge	23
3.3.12	eCommerce Transactions	24
4	USE CASES	25
4.1	RESERVATIONS.....	25
4.1.1	Reservation, Hold card number, No Authorization	25
4.1.2	Reservation, Hold card number, with Advance Deposit.....	25
4.2	CHANGE CARD.....	25
4.2.1	Change the card that is stored with the folio	25
4.3	CHECK-IN	26
4.3.1	Check in with the same card as reservation.....	26
4.3.2	Check in with no card presented	26
4.4	DURING STAY.....	27
4.4.1	Incremental authorization	27
4.5	CHECK-OUT.....	27
4.5.1	Incremental authorization	27
4.5.2	Reversal authorization	27
4.5.3	Check-out - transaction gets marked as "settleable".....	28
4.6	EXTENDED STAY SETTLEMENT	28
4.6.1	Transaction gets marked as "settleable"	28
4.7	CANCELLATION/NO-SHOW	28
4.7.1	Cancellation/No-show.....	28
4.8	BATCH CLOSE	29
4.8.1	Batch Close at end of day	29
4.9	VOID SETTLEMENT	29
4.9.1	Void (will remove a transaction from the current batch).....	29
4.10	RETURN	29
4.10.1	Return.....	29
4.11	POST-DEPARTURE CHARGE.....	29
4.11.1	Post-Departure Charge	29
4.12	E-COMMERCE TRANSACTIONS	30
4.12.1	Sale	30
5	PROCESSING REQUIREMENTS FOR BEST INTERCHANGE RATES- CARD BRAND SPECIFIC	31
6	MESSAGES	32

6.1.1	Request Data Element Table.....	32
6.1.2	Response Data Element Table	38
6.2	SAMPLE MESSAGES.....	42
6.2.1	Authorization.....	42
6.2.2	Incremental Authorization.....	43
6.2.3	Authorization Reversal.....	43
6.2.4	Settlement.....	44
6.2.5	Sale (No Prior Auth).....	45
6.2.6	Void Settlement (or Sale)	46
6.2.7	Return.....	47
7	PAYMENT CARD INDUSTRY DATA SECURITY STANDARD (PCI-DSS)	48
7.1	PCI-DSS	48
7.2	PA-DSS	48
7.3	RELATIONSHIP BETWEEN PCI DSS AND PA-DSS	48
7.4	TO WHICH APPLICATIONS DOES PA-DSS APPLY?.....	48
7.5	COMMUNICATIONS AND SECURITY	49

1 Document History

1.1 Document Changes

Version	Date	Author	Comments
0.1	26 Aug 2008	J. Newby	First Draft
0.2	10 Sep 2008	J. Newby	Second Draft with initial group comments
0.3	18 Sep 2008	S. Zloth	Third Draft
0.4	21 Oct 2008	S. Zloth	Fourth Draft
0.5	13 Nov 2008	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 Nov 2008	S. Zloth	Incorporated Common Classes, removed sections previously identified as ok to remove
0.7	12 Mar 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	1 Apr 2009	S. Zloth	Separated DataProxy into a different document. This document will be only for Payments functionality
0.9	10 Sep 2009	S. Zloth	Cleaned up doc for consistency with DataProxy spec.
1.2	23 Dec 2009	S. Zloth	Removed DCC, Multi-Currency and EMV (will be addressed in later version of the spec)
1.3	28 Jan 2010	J. Rosamilia	Added field defs and seeded samples for request and response messages.
1.4	1 Feb 2010	S. Zloth	Added Use Cases
1.5	3 Feb 2010	J. Rosamilia	Renamed messages and moved field def table.
1.6	11 Feb 2010	S. Zloth	Updated to include Business Process Flow diagrams
1.7	11 Feb 2010	J. Rosamilia	Added initial sample messages by transaction type.
1.8	12 Feb 2010	S. Zloth	Updated Roles and Use Cases
1.9	12 Feb 2010	S. Zloth	Added Implementation Requirements
0.9	12 Mar 2010		Versioning for member comment period; will lead to initial public release as version 1.0
0.9.1			
0.9.2	6 Apr 2010	J. Rosamilia	Implemented member comments.
1.0	23 Apr 2010		Public Release

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.

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 Process Flow, 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 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 Payment Cards.

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.
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 B ank I dentification N umber 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 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.
See below for specific card brand code names.	

	<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 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.
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	The organization or business providing goods or services to the cardholder.
MID	Merchant Identifier - the bank/acquirer allocated value to refer to the organization or business given the agreement to accept card transactions.
Multi-currency	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.
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
Payment Systems & Data Security Specifications 2010A, Data Proxy Specification Version 1.1	http://www.htng.org/specs/published.htm
OpenTravel Specifications	http://www.opentravel.org
PCI-DSS documentation	https://www.pcisecuritystandards.org/

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 hotel systems 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.

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 should 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 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.

3.3 Business Process Flow

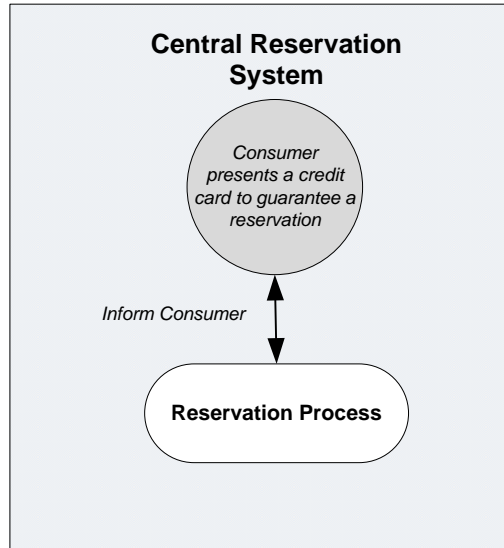
Authorization Request – payment data to be sent using the content and Web Service format as defined within this specification in order to obtain authorization from a card issuer for a stated amount. This Payment Authorization

Request may be sent by a Property Management System to a Payment Gateway (for one example) or by a Payment Gateway to the Payment Processing System (for another example).

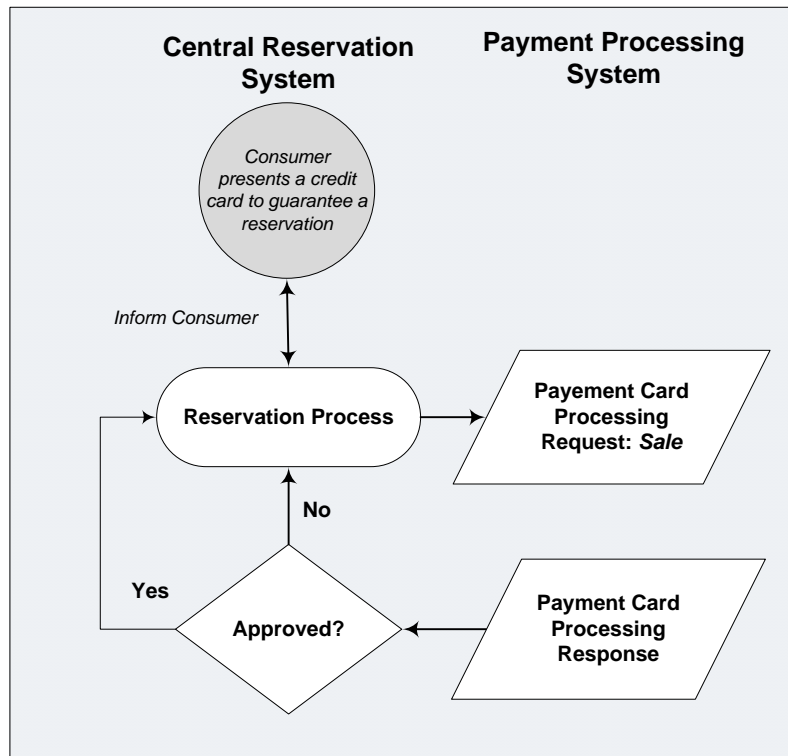
Authorization Response – payment data to be sent using the content and Web Service format as defined within this specification in order to convey response information from a card issuer for a stated amount. This Payment Authorization Response may be sent by a Payment Gateway to a Property Management System (for one example) or from the Payment Processing System to a Payment Gateway (for another example).

3.3.1 Reservation Process

3.3.1.1 Reservation made with guest providing card number but no authorization is secured.

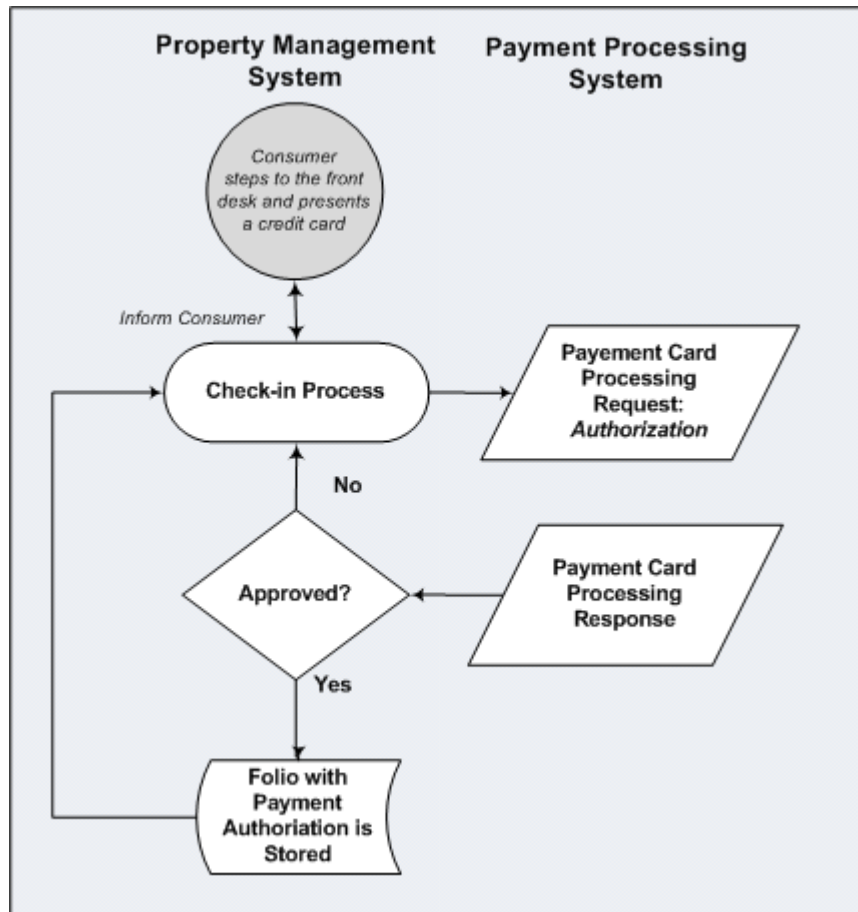


3.3.1.2 Reservation made with guest providing a card number. Payment is secured with the card and the transaction is settled.



3.3.3 Guest Check-in Process

3.3.3.1 Guest checks in with the same card used for the reservation

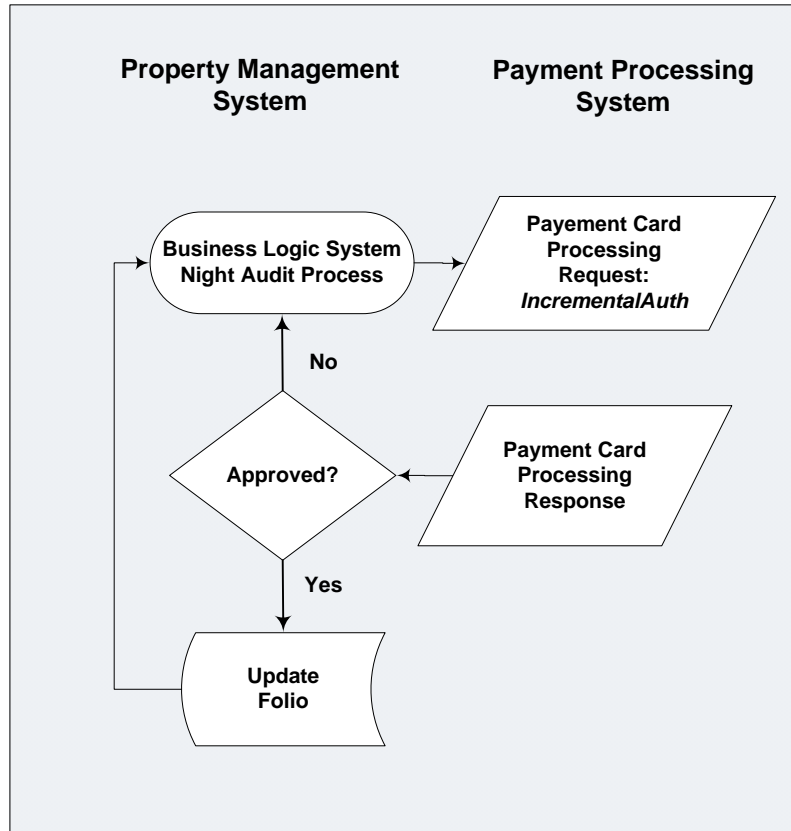


3.3.3.2 Guest checks in with no card, using a card-on-file instead

Use case diagram is the same as 3.3.3.1.

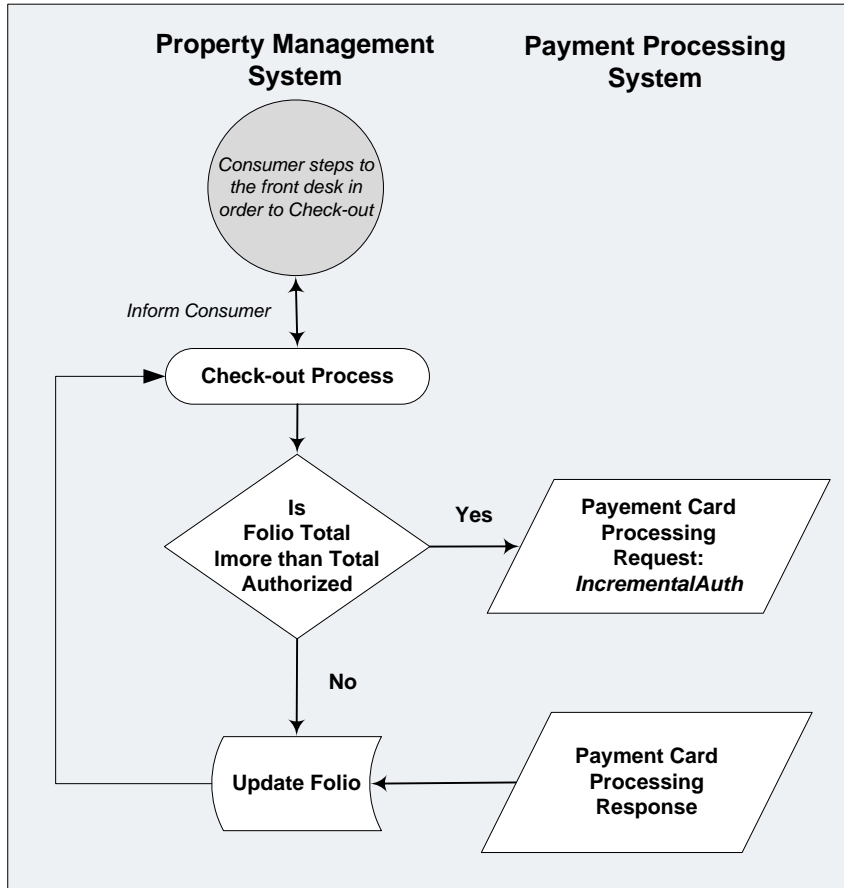
3.3.4 During Stay Activity

3.3.4.1 A checked in guest accumulates more charges than previously authorized. The Business Logic System recognizes this situation and obtains an incremental authorization.

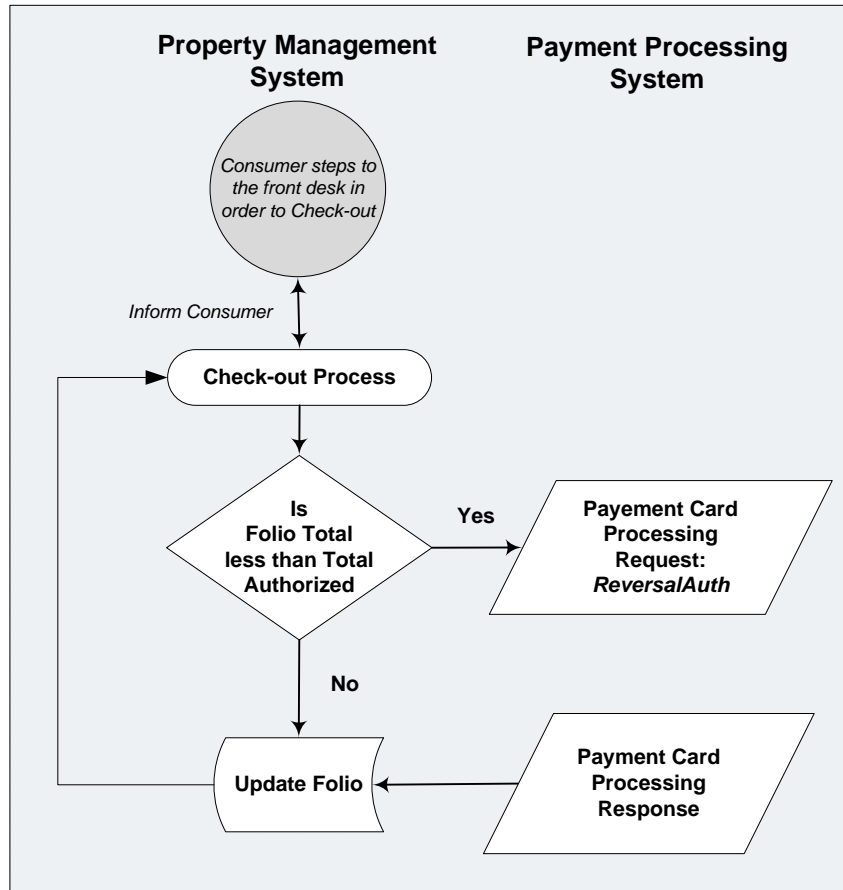


3.3.5 Guest Check-out Process

3.3.5.1 Guest is checking out. Guest's accumulated folio charges exceeds the amount currently authorized on the guest's card.

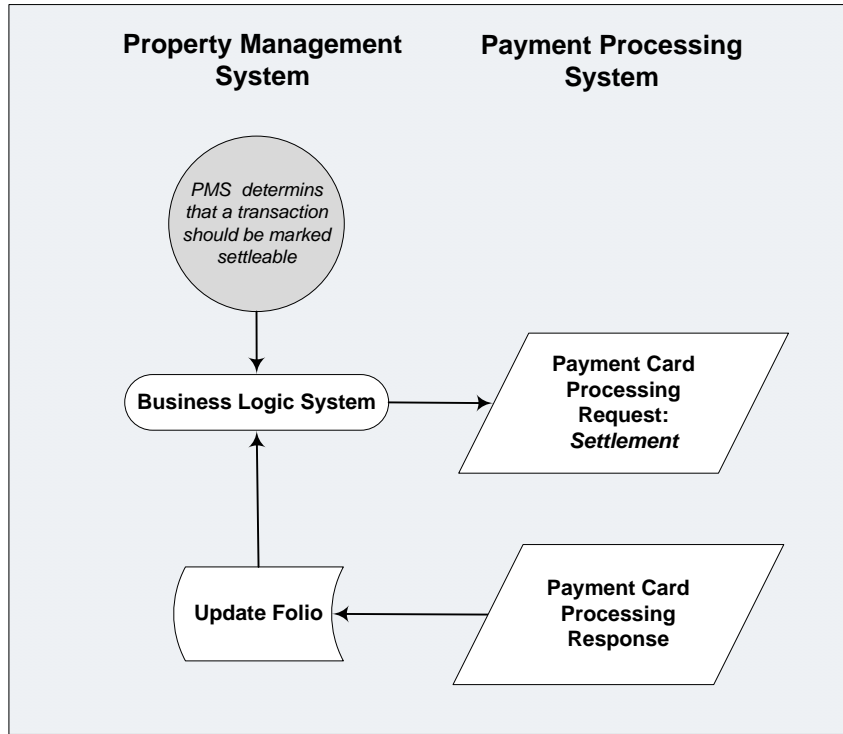


3.3.5.2 Guest is checking out. Guest's card has a higher amount authorized than accumulated in his/her folio.



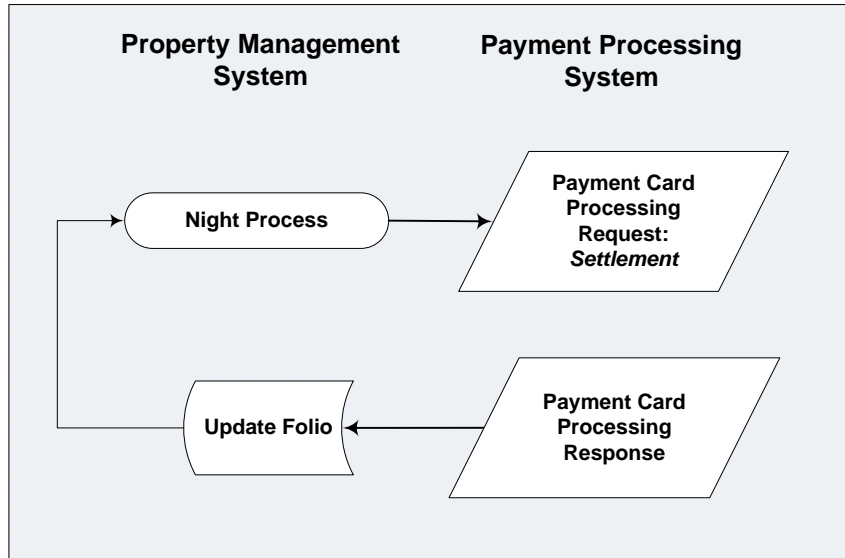
NOTE: Use case presumes that Folio Total cannot exceed Total Authorized as any attempt to accumulate a folio charge greater than the approved amount would result in an incremental authorization as illustrated in Use Case 3.1.

3.3.5.3 Upon completing transaction adjustments at checkout, the Business Logic System marks the Transaction as "settleable" for funds capture with next batch close process.



3.3.6 Extended Stay Processing

3.3.6.1 During a guests extended stay, the Business Logic System marks a guest current transaction as settleable for funds capture.



3.3.7 Cancellation/No-show

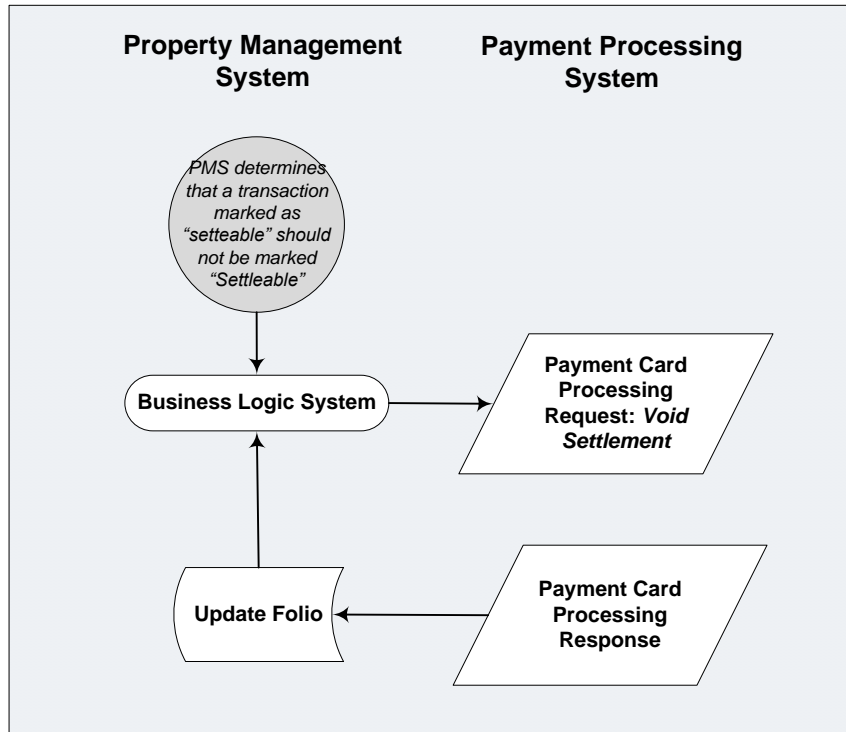
3.3.7.1 Cancellation/No-show use cases are the same as Check-out – no additional Use Cases needed.

3.3.8 Batch Close

Currently, this specification does not support a batch close process. Refer to 4.8.1 for more details.

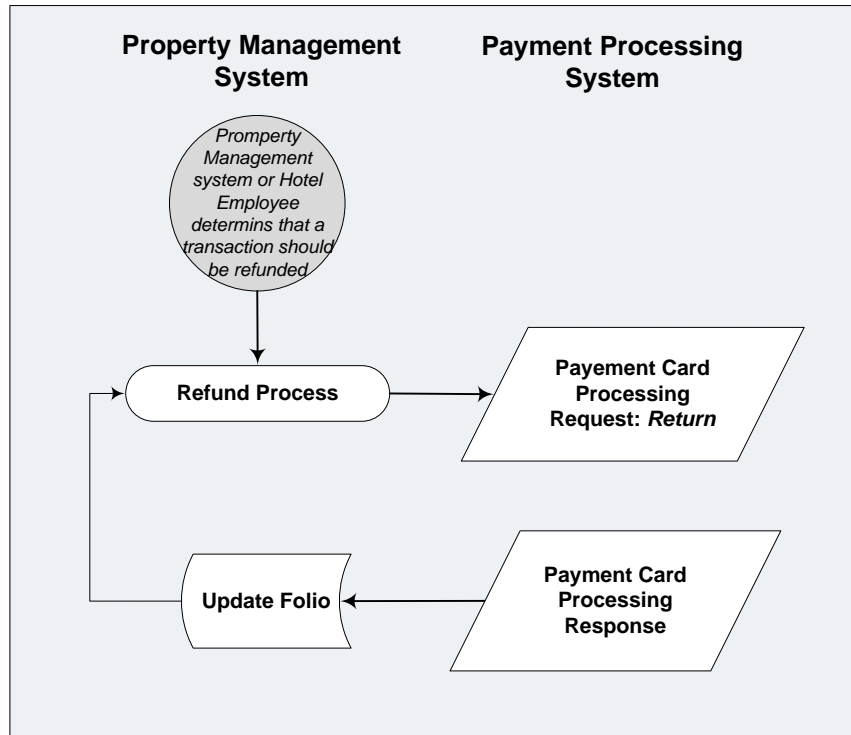
3.3.9 Void Settlement, marking a transaction as not "settleable"

3.3.9.1 Since the payment also exists in the Payment Processing System, the Business Logic System should invoke a Void Settlement Use Case (see 4.9) to suspend or cancel a previous request to settle a transaction.



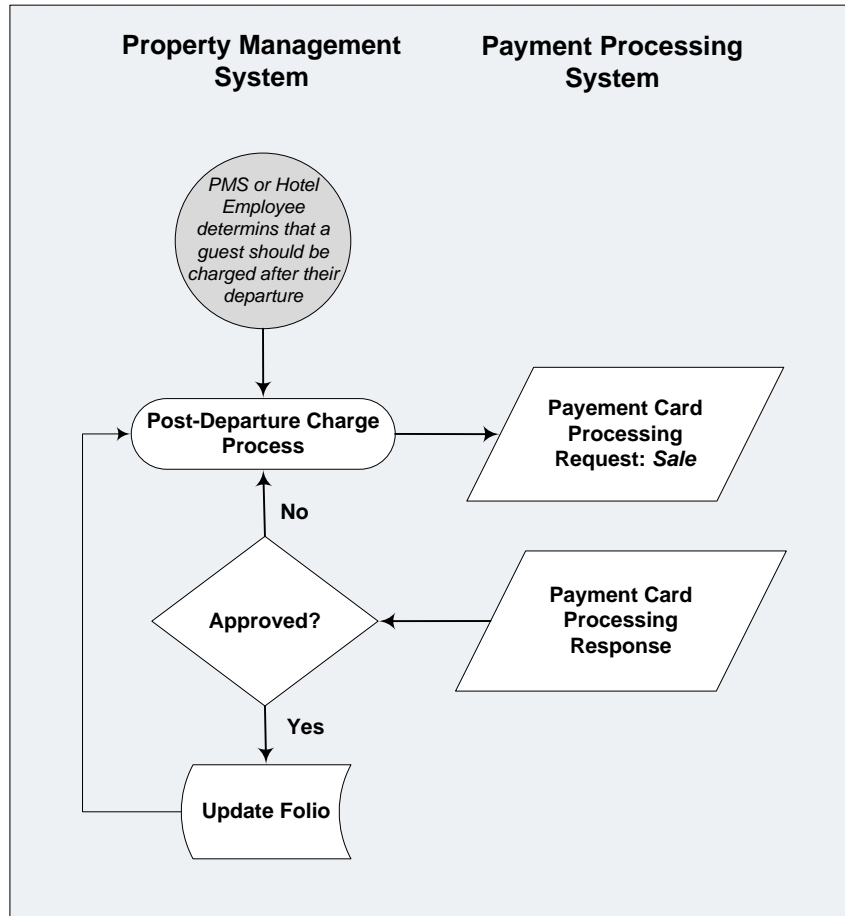
3.3.10 Return Transaction

3.3.10.1 Property Management System or Hotel personnel determine that a return transaction is required.



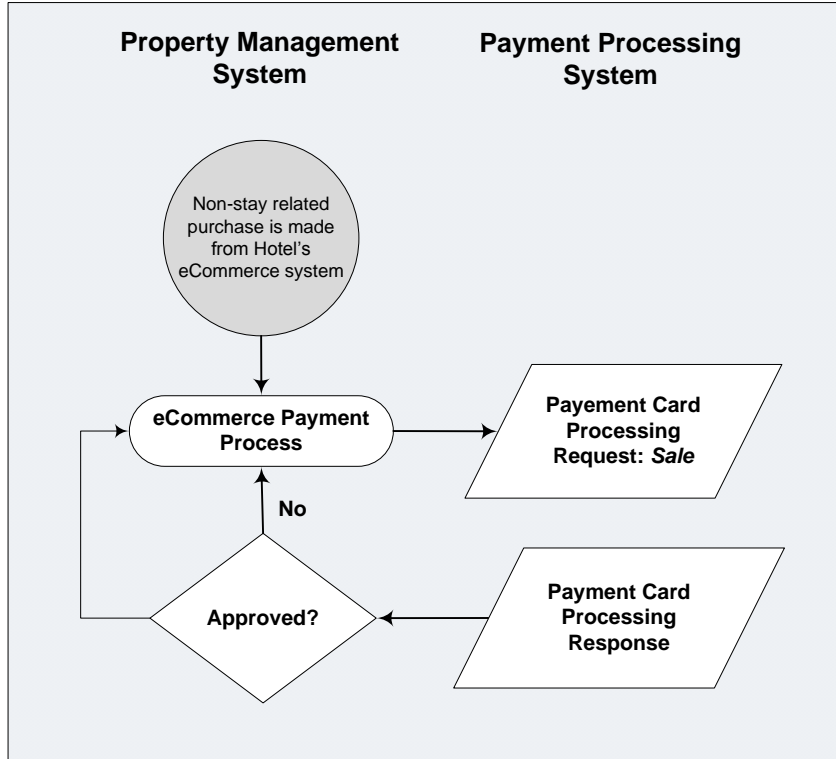
3.3.11 Post Departure Charge

3.3.11.1 After a guest checks out, it is discovered that a charge needs to be applied to the guest's card.



3.3.12 eCommerce Transactions

3.3.12.1 A purchase is made from the property's eCommerce site that is not related to a stay at the hotel (refer to 4.12)



4 Use Cases

Transaction Types

- HTNG_PaymentCardProcessingRQ
- HTNG_PaymentCardProcessingRS
- HTNG_PaymentCardProxyRQ (detailed in HTNG Payment Systems and Data Security 2009B Specification)
- HTNG_PaymentCardProxyRS (detailed in HTNG Payment Systems and Data Security 2009B Specification)

4.1 Reservations

4.1.1 Reservation, Hold card number, No Authorization

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- Guest makes a reservation and the card number is exchanged for a DataProxy.

Basic Flow

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

Preconditions

- None

Postconditions

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

4.1.2 Reservation, Hold card number, with Advance Deposit

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- Guest makes a reservation and the card number is exchanged for a DataProxy, and the card is authorized and charged for a specified amount.

Basic Flow

- The use case starts when the actor identifies that a guest is entering a payment card.
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Sale) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- Note: the action to exchange a payment card for a DataProxy can be part of the HTNG_PaymentCardProcessingRQ message (if the Payment Processing System is acting as the Proxy Vault), or the Actor can issue a separate HTNG_PaymentCardProxyRQ to the Proxy Vault (if the Payment Processing System is a separate role from the Proxy Vault).
- The use case terminates.

Preconditions

- None

Postconditions

- The DataProxy is stored with the guest's reservation record as a replacement for the payment card number. The DataProxy and authorization/payment information are stored in the PMS.

4.2 Change Card

4.2.1 Change the card that is stored with the folio

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- Guest changes the card he is using. This can take place during the stay. The authorization on the original card should be reversed.
- Basic Flow
- The use case starts when the actor identifies that a guest is entering a payment card.
 - The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Authorization) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
 - The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to AuthReversal) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
 - The use case terminates.
- Preconditions
- None
- Postconditions
- The DataProxy is stored with the guest’s reservation record as a replacement for the payment card number. The DataProxy and authorization information are stored in the PMS.

4.3 Check-in

4.3.1 Check in with the same card as reservation

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

- Brief Description
- Guest checks in to the hotel. A card is swiped and an authorization request is sent for the estimated dollar amount of the guest’s stay.
- Basic Flow
- The use case starts when the actor identifies that a guest is entering a payment card.
 - The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Authorization) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return. We recommend this request include the track data from the swipe card.
 - Note: the action to exchange a payment card for a DataProxy can be part of the HTNG_PaymentCardProcessingRQ message (if the Payment Processing System is acting as the Proxy Vault), or the Actor can issue a separate HTNG_PaymentCardProxyRQ to the Proxy Vault (if the Payment Processing System is a separate role from the Proxy Vault).
 - The use case terminates.
- Preconditions
- None
- Postconditions
- The DataProxy is stored with the guest’s reservation record as a replacement for the payment card number. The DataProxy and authorization information are stored in the PMS.

4.3.2 Check in with no card presented

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

- Brief Description
- Guest checks in to the hotel. No card is swiped. An authorization request is sent using the DataProxy or Card Number on file for the reservation for the estimated dollar amount of the guest’s stay.
- Basic Flow
- The use case starts when the actor identifies that a guest is entering a payment card.
 - The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Authorization) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
 - Note: the action to exchange a payment card for a DataProxy can be part of the HTNG_PaymentCardProcessingRQ message (if the Payment Processing System is acting as the Proxy Vault), or the Actor can issue a separate HTNG_PaymentCardProxyRQ to the Proxy Vault (if the Payment Processing System is a separate role from the Proxy Vault).
 - The use case terminates.
- Preconditions
- None

Postconditions

- The DataProxy is stored with the guest's reservation record as a replacement for the payment card number. The DataProxy and authorization information are stored in the PMS.

4.4 During stay

4.4.1 Incremental authorization

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- A guest is checked in. The guest accumulates more charges than there is authorizations for on his/her folio.

Basic Flow

- The use case starts when the Incremental Authorization System recognizes that additional authorization is needed.
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to IncrementalAuthorization) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- The use case terminates.

Preconditions

- None

Postconditions

- The authorization information is stored in the PMS.

4.5 Check-out

4.5.1 Incremental authorization

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- A guest is checking out. The guest has a lower amount authorized than accumulated on his/her folio.

Basic Flow

- The use case starts when the Incremental Authorization System recognizes that there is too little in authorizations on the folio.
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to IncrementalAuthorization) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.

Preconditions

- None

Postconditions

- The authorization information is stored in the PMS.

4.5.2 Reversal authorization

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- A guest is checking out. The guest has a higher amount authorized than accumulated on his/her folio.

Basic Flow

- The use case starts when the Reversal Authorization System recognizes that there is too much in authorizations on the folio.
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to AuthReversal) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return. The use case terminates.

Preconditions

- None

Postconditions

- The authorization information is stored in the PMS.

4.5.3 Check-out - transaction gets marked as "settleable"

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- A guest has checked out. The transaction is sent to the Payment Processing System for later settlement.
- The Actor issues an HTNG_PaymentCardProcessingRQ to the Payment Processing System (with the TransactionType set to Settlement) and will receive an HTNG_PaymentCardProcessingRS in return.
- The use case terminates.

Preconditions

- None

Postconditions

- The transaction is marked as Settled in the PMS and Settleable in the Payment Processing System.

4.6 Extended Stay Settlement

4.6.1 Transaction gets marked as "settleable"

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- The folio needs to be settled while the guest is still checked in. The transaction is sent to the Payment Processing System for later settlement.
- The Actor issues an HTNG_PaymentCardProcessingRQ to the Payment Processing System (with the TransactionType set to Settlement) and will receive an HTNG_PaymentCardProcessingRS in return.
- The use case terminates.

Preconditions

- None

Postconditions

- The transaction is marked as Settled in the PMS.

4.7 Cancellation/No-show

4.7.1 Cancellation/No-show

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- Guest makes a reservation and the card number is exchanged for a DataProxy. The guest does not show up and the merchant needs to charge the card for a specified amount.

Basic Flow

- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Sale) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- Note: the action to exchange a payment card for a DataProxy can be part of the HTNG_PaymentCardProcessingRQ message (if the Payment Processing System is acting as the Proxy Vault), or the Actor can issue a separate HTNG_PaymentCardProxyRQ to the Proxy Vault (if the Payment Processing System is a separate role from the Proxy Vault).
- The use case terminates.

Preconditions

- None

Postconditions

- The DataProxy is stored with the guest's reservation record as a replacement for the payment card number. The DataProxy and authorization/payment information are stored in the PMS.

4.8 Batch Close

4.8.1 Batch Close at end of day

No use case necessary – no transaction sent to the Payment Processing System.

- The assumption is that the user will either initiate settlement directly on the Payment Processing System (via a web interface) or the Payment Processing System will “auto-settle” the transactions at a specified time.

4.9 Void Settlement

4.9.1 Void (will remove a transaction from the current batch)

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- The settleable status of a transaction from the current day is reversed. The transaction remains available for additional incremental authorizations or reversal authorizations requests if appropriate.

Basic Flow

- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to VoidSettlement) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.

Preconditions

- None

Postconditions

- None

4.10 Return

4.10.1 Return

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- A Credit or Return (negative charge) needs to be applied to a credit card.

Basic Flow

- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Return) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- The use case terminates.

Preconditions

- None

Postconditions

- The Credit information is stored in the PMS.

4.11 Post-Departure Charge

4.11.1 Post-Departure Charge

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- A charge needs to be applied to a credit card after the guest has checked out.

Basic Flow

- The Actor issues a HTNG_PaymentCardProcessingRQ (with the TransactionType set to Sale) to the Payment Processing System and will receive a HTNG_PaymentCardProcessingRS in return.
- The use case terminates.

Preconditions

- None

Postconditions

- The charge information is stored in the PMS. It will be settled with the current day's batch.

4.12 eCommerce Transactions

4.12.1 Sale

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- Purchase is made on the hotel's eCommerce site. The purchase is not related to a stay at the hotel.

Basic Flow

- The use case starts when the actor identifies that a guest is entering a payment card.
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Sale) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- Note: the action to exchange a payment card for a DataProxy can be part of the HTNG_PaymentCardProcessingRQ message (if the Payment Processing System is acting as the Proxy Vault), or the Actor can issue a separate HTNG_PaymentCardProxyRQ to the Proxy Vault (if the Payment Processing System is a separate role from the Proxy Vault).
- The use case terminates.

Preconditions

- None

Postconditions

- The DataProxy is stored with the purchase record as a replacement for the payment card number.

5 Processing requirements for best interchange rates– Card Brand Specific

Below are the Card Association requirements for the best interchange rates. Please note that these are the requirements as of September 2008, and they may change over time. Integrators should consult with industry sources to get up-to-date information.

United States:

Visa

- The transaction must be settled within 48 hours after the check-out.
- The check-out amount must be within 15%** of the original authorization amount. (One Auth Reversal and multiple Incremental Auths are allowed to bring the authorization amount within tolerance.)
- The transaction date must match the check-out date.
- The original approval number must be submitted in settlement.

MasterCard

- The transaction must be settled within 24 hours after the check-out.
- The transaction date must match the check-out date.

American Express

- The check-in and check-out dates must be valid dates.
- The check-in date must be less than the check-out date.

6 Messages

6.1.1 Request Data Element Table

Element @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
HTNG_PaymentCardProcessingRQ	A generic message, available as an action on several OpenTravel services which requests a server to read and return the document type identified by the UniqueID element.	Req	Req	Req	Req	Req	Req	Req
@EchoToken	A reference for additional message identification, assigned by the requesting host system. The corresponding response message MUST include an echo token with an identical value.	Req	Req	Req	Req	Req	Req	Req
@TransactionIdentifier	Uniquely identifies a given transaction chain. This is issued by the Payment Gateway when a new transaction chain is started. Subsequent transactions in the chain should reference this value.	N/A	Req	Req	Req	N/A	Req	N/A
@TimeStamp	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24 hour clock (e.g. 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).	Req	Req	Req	Req	Req	Req	Req
@Version	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.	Req	Req	Req	Req	Req	Req	Req
@Target	Used to indicate whether the request is for the Test or Production system.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRQ / POS / Source	This holds details regarding the requestor. It may be repeated to also accommodate the delivery systems.	Req	Req	Req	Req	Req	Req	Req

Element @Attribute	Description/Contents	Authorizatio n	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
@TerminalID	This is the electronic address of the device from which information is entered.	Req	Req	Req	Req	Req	Req	Req
HTNG_PaymentC ardProcessingRQ / POS / Source / RequestorID	An identifier of the entity making the request (e.g. ATA/IATA/ID number, Electronic Reservation Service Provider (ERSP), Association of British Travel Agents (ABTA)).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Type	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ID	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.	Req	Req	Req	Req	Req	Req	Req
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail		Req	Req	Req	Req	Req	Req	Req
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation	Specifies credit card information about the customer seeking authorization.	Req	Req	Req	Req	Req	Req	Req
@TransactionTyp e	Authorization, Incremental Authorization, Auth Reversal, Settlement, Sale, Void Settlement, Return	Req	Req	Req	Req	Req	Req	Req
@SaleCode	Advance Deposit, Delayed Charge, Express Service, Assured Reservation, Normal Charge, No Show Charge	Req	Opt	Req	Req	Req	Opt	Opt
@AuthorizationCo de	This is the approval code received on the original authorization request. Only used in the case where the requested transaction is to reverse the authorization.	N/A	N/A	N/A	Req	N/A	N/A	N/A
@CardPresentInd	When true, the customer has actually presented the credit card.	Req	Req	Req	Req	Req	Req	Req

Element @Attribute	Description/Contents	Authorizatio n	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
@Amount	A monetary amount.	Req	Req	Req	Req	Req	Req	Req
@AuthVerification Value	The cardholder authentication verification value required for some credit card authorization, such as the Verified by Visa Process.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@SourceType	Information describing the point of sale. Enumeration: NormalTransaction, MailOrPhoneOrder, UnattendedTerminal, MerchantIsSuspicious, eCommerceSecuredTransaction, eCommerceUnsecuredTransaction, InFlightAirPhone,CIS_Not Legible, CID_NotOnCard	Req	Req	Req	Req	Req	Req	Req
@E_CommerceCode	The electronic commerce indicator required for some credit card authorizations, such as the Verified by Visa Process.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard	Specifies the credit card information for which authorization is required.	Req	Req	Req	Req	Req	Req	Req
@CardNumber	Credit card number embossed on the card.	Cond Req	Cond Req	Cond Req	Cond Req	Cond Req	Cond Req	Cond Req
@CardNumberIsProxy		Opt	Opt	Opt	Opt	Opt	Opt	Opt
@MaskedCardNumber	May be used to send a concealed credit card number (e.g., xxxxxxxxxxxx9922).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@CardCode	The 2 character code of the credit card issuer.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ExpireDate	Indicates the ending date.	Cond Req	N/A	N/A	Cond Req	Cond Req	N/A	Cond Req
@SeriesCode	Verification digits printed on the card following the embossed number. This may also accommodate the customer identification/batch number (CID), card verification value (CVV2), card validation code number (CVC2) on credit card.	Cond Req	N/A	N/A	Cond Req	Cond Req	N/A	Cond Req

Element @Attribute	Description/Contents	Authorizatio n	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / CardHolderName	Name of the card holder.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / Address	Card holder's address used for additional authorization checks. Should be billing address of card holder.	Cond Req	Opt	Opt	Opt	Cond Req	Opt	Opt
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / Address / AddressLine	May contain the street number and optionally the street name.	Cond Req	Opt	Opt	Opt	Cond Req	Opt	Opt
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / Address / CityName	City (e.g., Dublin), town, or postal station (i.e., a postal service territory, often used in a military address).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / Address / PostalCode	Post Office Code number.	Cond Req	Opt	Opt	Opt	Cond Req	Opt	Opt
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / Address / StateProv	State or Province name (e.g., Texas).	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element @Attribute	Description/Contents	Authorizatio n	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / Address / CountryName	Country name (e.g., Ireland).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / CreditCard / MagneticStripe	Card Magnetic Stripe Data as defined by ISO 7813 for banking cards.	Cond Req	Opt	Opt	Opt	Opt	Opt	Opt
@Track1	The binary magnetic stripe data for track 1.	Cond Req	N/A	N/A	Opt	Opt	Opt	Opt
@Track2	The binary magnetic stripe data for track 2.	Cond Req	N/A	N/A	Opt	Opt	Opt	Opt
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / ID	Identification of an authorization party (e.g., merchant, acquirer).	Req	Req	Req	Req	Req	Req	Req
@Type	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ID	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.	Req	Req	Req	Req	Req	Req	Req
HTNG_PaymentC ardProcessingRQ / AuthorizationDet ail / CreditCardAuthori zation / ID / CompanyName	Identifies the company that is associated with the UniqueID.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@CompanyShort Name	Used to provide the company common name.	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element @Attribute	Description/Contents	Authorizatio n	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
HTNG_PaymentCardProcessingRQ / StayInfo / RevenueCategories / RevenueCategory	The classifications of revenue data associated with the StayInfo report. A RevenueCategory provide a way to classify guest financial stay data and analyze guest spending for a certain category (e.g., food and beverage, room, etc.)	Opt	Opt	Opt	Req	Req	Req	Opt
@RevenueCategoryCode	Describes the type of revenue generated. Refer to OpenTravel Code List Revenue Category Code (RCC).	Opt	Opt	Opt	Req	Req	Req	Opt
HTNG_PaymentCardProcessingRQ / StayInfo / RevenueCategories / RevenueCategory / SummaryAmount		Opt	Opt	Opt	Req	Req	Req	Opt
@Amount	A monetary amount.	Opt	Opt	Opt	Req	Req	Req	Opt
HTNG_PaymentCardProcessingRQ / StayInfo / ReservationID	The confirmation number of the reservation associated with the stay.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Type	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT). This should be able to be tracked back to an individual reservation or folio.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ID	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.	Req	Req	Req	Req	Req	Req	Req
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRQ / StayInfo / HotelReservation / RoomStays / RoomStay / TimeSpan	The Time Span which covers the Room Stay.	Req	Req	Req	Req	Req	Req	Req
@End	The ending value of the time span. (e.g. Departure Date)	Req	Req	Req	Req	Req	Req	Req

Element @Attribute	Description/Contents	Authorizatio n	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
@Start	The starting value of the time span. (e.g. Arrival Date)	Req	Req	Req	Req	Req	Req	Req

6.1.2 Response Data Element Table

Element @Attribute	Description/Contents	Authorizatio n	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
HTNG_Payment CardProcessingR S	A generic message, available as an action on several OpenTravel services which requests a server to read and return the document type identified by the UniqueID element.	Req	Req	Req	Req	Req	Req	Req
@EchoToken	A reference for additional message identification, assigned by the requesting host system. The corresponding response message MUST include an echo token with an identical value.	Req	Req	Req	Req	Req	Req	Req
@TransactionIde ntifier	Uniquely identifies a given transaction chain. This is issued by the Payment Gateway when a new transaciton chain is started. Subsequent transactions in the chain should reference this value.	Req	Req	Req	Req	Req	Req	Req
@TimeStamp	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24 hour clock (e.g. 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).	Req	Req	Req	Req	Req	Req	Req
@Version	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.	Req	Req	Req	Req	Req	Req	Req
@Target	Used to indicate whether the request is for the Test	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
	or Production system.							
HTNG_Payment CardProcessingR S / Success	Standard way to indicate successful processing of an OpenTravel message. Returning an empty element of this type indicates success.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_Payment CardProcessingR S / Warnings / Warning	Used in conjunction with the Success element to define a business error.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Type	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 OpenTravel Code List Error Warning Type (EWT).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Status	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.	Req	Req	Req	Req	Req	Req	Req
@ShortText	An abbreviated version of the error in textual format.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Code	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_Payment CardProcessingR S / Authorization / AuthorizationDet ail	The original authorization request information.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_Payment CardProcessingR S /	Specifies credit card information about the customer seeking	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
Authorization / AuthorizationDetail / CreditCardAuthorization	authorization.							
@TransactionType	Authorization, Incremental Authorization, Auth Reversal, Settlement, Sale, Void Settlement, Return	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@SaleCode	Advance Deposit, Delayed Charge, Express Service, Assured Reservation, Normal Charge, No Show Charge	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@AuthorizationCode	This is the approval code received on the original authorization request. Only used in the case where the requested transaction is to reverse the authorization.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Amount	A monetary amount.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_Payment CardProcessingRS / Authorization / AuthorizationDetail / CreditCardAuthorization / CreditCard	Specifies the credit card information for which authorization is required.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@MaskedCardNumber	May be used to send a concealed credit card number (e.g., xxxxxxxxxxx9922).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@CardCode	The 2 character code of the credit card issuer.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ExpireDate	Indicates the ending date.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_Payment CardProcessingRS / Authorization / AuthorizationDetail / CreditCardAuthorization / ID	Identification of an authorization party (e.g., merchant, acquirer).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Type	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ID	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
	particular implementation.							
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_Payment CardProcessingR S / Authorization / AuthorizationDet ail / BookingReferenc eID	The booking or confirmation number for which charges are being authorized.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Type	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ID	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_Payment CardProcessingR S / Authorization / AuthorizationRes ult	Result information from the authorization process.	Req	Req	Req	Req	Req	Req	Req
@ApprovalDateT ime	The date and time that the approval was issued.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@CVC_Result	A response from the validation of the Card Verification Code (CVC/CSC- Card Security Code or CVV/CVS Card Verification Value), a security feature for credit card transactions.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@AddressResult Code	The result of the address validation.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@AuthorizationC ode	The unique code returned by the authorizing party. This is returned for successful authorizations.	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@Result	Information returned by the credit card vendor describing the results of the processing of the request.	Req	Req	Req	Req	Req	Req	Req

Element @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return
@Description	Additional response information.	Req	Req	Req	Req	Req	Req	Req

6.2 Sample Messages

6.2.1 Authorization

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRQ EchoToken="200f9d19-f621-443f-adce-2c7de7c6afe2" TimeStamp="2010-02-11T16:35:59" Version="1.0" xmlns="http://htng.org/2010A"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <POS>
    <Source TerminalID="192.168.12.87"/>
  </POS>
  <AuthorizationDetail>
    <CreditCardAuthorization TransactionType="Authorization" SaleCode="Normal Charge"
CardPresentInd="true" Amount="467.36" SourceType="NormalTransaction">
      <CreditCard CardNumber="412345678901234" ExpireDate="0212"
SeriesCode="123"/>
      <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>
    </CreditCardAuthorization>
  </AuthorizationDetail>
  <StayInfo>
    <ReservationID Type="14" ID_Context="PMSABC" ID="567841"/>
    <HotelReservation>
      <RoomStays>
        <RoomStay>
          <TimeSpan End="2010-03-01" Start="2010-03-04"/>
        </RoomStay>
      </RoomStays>
    </HotelReservation>
  </StayInfo>
</HTNG_PaymentCardProcessingRQ>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRS EchoToken="200f9d19-f621-443f-adce-2c7de7c6afe2"
TransactionIdentifier="68473921" TimeStamp="2010-02-11T16:36:06" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Success/>
  <Authorization>
    <AuthorizationDetail>
      <CreditCardAuthorization TransactionType="Authorization" SaleCode="Normal
Charge" AuthorizationCode="1215" Amount="467.36">
        <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" CardCode="VI"
ExpireDate="0212"/>
        <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>
      </CreditCardAuthorization>
      <BookingReferenceID Type="14" ID_Context="PMSABC" ID="567841" />
    </AuthorizationDetail>
    <AuthorizationResult ApprovalDateTime="2010-02-11T16:36:03" CVC_Result="Match"
AddressResultCode="Z" AuthorizationCode="1215" Result="Approved" Description="Authorization
successful"/>
  </Authorization>
</HTNG_PaymentCardProcessingRS>
```

6.2.2 Incremental Authorization

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRQ EchoToken="fc52b38a-1288-454e-898c-6956c68cdac6"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:27" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <POS>
    <Source TerminalID="192.168.12.79"/>
  </POS>
  <AuthorizationDetail>
    <CreditCardAuthorization TransactionType="Incremental Authorization"
SaleCode="Normal Charge" CardPresentInd="true" Amount="58.92" SourceType="NormalTransaction">
      <CreditCard CardNumber="412345678901234" ExpireDate="0212"/>
      <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>
    </CreditCardAuthorization>
  </AuthorizationDetail>
  <StayInfo>
    <ReservationID Type="14" ID_Context="PMSABC" ID="567841"/>
    <HotelReservation>
      <RoomStays>
        <RoomStay>
          <TimeSpan End="2010-03-01" Start="2010-03-04"/>
        </RoomStay>
      </RoomStays>
    </HotelReservation>
  </StayInfo>
</HTNG_PaymentCardProcessingRQ>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRS EchoToken="fc52b38a-1288-454e-898c-6956c68cdac6"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:53" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Success/>
  <Authorization>
    <AuthorizationDetail>
      <CreditCardAuthorization TransactionType="Incremental Authorization"
SaleCode="Normal Charge" AuthorizationCode="1215" Amount="58.92">
        <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" CardCode="VI"
ExpireDate="0212"/>
        <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>
      </CreditCardAuthorization>
      <BookingReferenceID Type="14" ID_Context="PMSABC" ID="567841" />
    </AuthorizationDetail>
    <AuthorizationResult ApprovalDateTime="2010-02-12T01:21:35"
AuthorizationCode="1215" Result="Approved" Description="Incremental Authorization successful"/>
  </Authorization>
</HTNG_PaymentCardProcessingRS>
```

6.2.3 Authorization Reversal

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRQ EchoToken="ea2520b4-4738-4d87-880a-a9bccd7988a9"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:57" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <POS>
    <Source TerminalID="192.168.12.82"/>
  </POS>
  <AuthorizationDetail>
    <CreditCardAuthorization TransactionType="Auth Reversal" SaleCode="Normal Charge"
CardPresentInd="true" Amount="526.28" SourceType="NormalTransaction">
      <CreditCard CardNumber="412345678901234" ExpireDate="0212"/>
      <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>
    </CreditCardAuthorization>
  </AuthorizationDetail>
  <StayInfo>
```

```
<ReservationID Type="14" ID_Context="PMSABC" ID="567841"/>
<HotelReservation>
  <RoomStays>
    <RoomStay>
      <TimeSpan End="2010-03-01" Start="2010-03-04"/>
    </RoomStay>
  </RoomStays>
</HotelReservation>
</StayInfo>
</HTNG_PaymentCardProcessingRQ>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRS EchoToken="ea2520b4-4738-4d87-880a-a9bccd7988a9"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:03" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Success/>
  <Authorization>
    <AuthorizationDetail>
      <CreditCardAuthorization TransactionType="Auth Reversal" SaleCode="Normal
Charge" AuthorizationCode="1215" Amount="526.28">
        <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" CardCode="VI"
ExpireDate="0212"/>
        <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>
      </CreditCardAuthorization>
      <BookingReferenceID Type="14" ID_Context="PMSABC" ID="567841" />
    </AuthorizationDetail>
    <AuthorizationResult ApprovalDateTime="2010-02-12T01:21:25"
AuthorizationCode="1215" Result="Approved" Description="Auth Reversal successful"/>
  </Authorization>
</HTNG_PaymentCardProcessingRS>
```

6.2.4 Settlement

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRQ EchoToken="a11cd3ae-c4fa-4710-abf2-56957a9714a9"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:07" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <POS>
    <Source TerminalID="192.168.12.81"/>
  </POS>
  <AuthorizationDetail>
    <CreditCardAuthorization TransactionType="Settlement" SaleCode="Normal Charge"
CardPresentInd="true" Amount="526.28" SourceType="NormalTransaction">
      <CreditCard CardNumber="412345678901234" ExpireDate="0212"/>
      <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>
    </CreditCardAuthorization>
  </AuthorizationDetail>
  <StayInfo>
    <RevenueCategories>
      <RevenueCategory RevenueCategoryCode="10">
        <SummaryAmount Amount="400.00"/>
      </RevenueCategory>
      <RevenueCategory RevenueCategoryCode="2">
        <SummaryAmount Amount="100.00"/>
      </RevenueCategory>
      <RevenueCategory RevenueCategoryCode="14">
        <SummaryAmount Amount="10.00"/>
      </RevenueCategory>
    </RevenueCategories>
    <ReservationID Type="14" ID_Context="PMSABC" ID="567841"/>
    <HotelReservation>
      <RoomStays>
        <RoomStay>
          <TimeSpan End="2010-03-01" Start="2010-03-04"/>
        </RoomStay>
      </RoomStays>
    </HotelReservation>
  </StayInfo>
</HTNG_PaymentCardProcessingRQ>
```

```
</HotelReservation>  
</StayInfo>  
</HTNG_PaymentCardProcessingRQ>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<HTNG_PaymentCardProcessingRS EchoToken="a11cd3ae-c4fa-4710-abf2-56957a9714a9"  
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:13" Version="1.0"  
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  <Success/>  
  <Authorization>  
    <AuthorizationDetail>  
      <CreditCardAuthorization TransactionType="Settlement" SaleCode="Normal  
Charge" AuthorizationCode="1215" Amount="526.28">  
        <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" CardCode="VI"  
ExpireDate="0212"/>  
        <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>  
      </CreditCardAuthorization>  
      <BookingReferenceID Type="14" ID_Context="PMSABC" ID="567841" />  
    </AuthorizationDetail>  
    <AuthorizationResult ApprovalDateTime="2010-02-12T01:21:15"  
AuthorizationCode="1215" Result="Approved" Description="Settlement successful"/>  
  </Authorization>  
</HTNG_PaymentCardProcessingRS>
```

6.2.5 Sale (No Prior Auth)

Request

```
<?xml version="1.0" encoding="UTF-8"?>  
<HTNG_PaymentCardProcessingRQ EchoToken="d328b365-d896-4914-aab5-2b6f204ae82f" TimeStamp="2010-02-  
11T16:35:59" Version="1.0" xmlns="http://htng.org/2010A"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  <POS>  
    <Source TerminalID="192.168.12.87"/>  
  </POS>  
  <AuthorizationDetail>  
    <CreditCardAuthorization TransactionType="Sale" SaleCode="No Show Charge"  
CardPresentInd="false" Amount="467.36" SourceType="NormalTransaction">  
      <CreditCard CardNumber="412345678901234" ExpireDate="0212">  
        <Address>  
          <AddressLine>101 Main Street</AddressLine>  
          <CityName>Anytown</CityName>  
          <StateProv>PA</StateProv>  
          <PostalCode>01234</PostalCode>  
        </Address>  
      </CreditCard>  
      <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>  
    </CreditCardAuthorization>  
  </AuthorizationDetail>  
  <StayInfo>  
    <RevenueCategories>  
      <RevenueCategory RevenueCategoryCode="10">  
        <SummaryAmount Amount="400.00"/>  
      </RevenueCategory>  
      <RevenueCategory RevenueCategoryCode="2">  
        <SummaryAmount Amount="100.00"/>  
      </RevenueCategory>  
      <RevenueCategory RevenueCategoryCode="14">  
        <SummaryAmount Amount="10.00"/>  
      </RevenueCategory>  
    </RevenueCategories>  
    <ReservationID Type="14" ID_Context="PMSABC" ID="567841"/>  
  <HotelReservation>  
    <RoomStays>  
      <RoomStay>  
        <TimeSpan End="2010-03-01" Start="2010-03-04"/>  
      </RoomStay>  
    </RoomStays>
```

```
        </HotelReservation>
    </StayInfo>
</HTNG_PaymentCardProcessingRQ>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRS EchoToken="d328b365-d896-4914-aab5-2b6f204ae82f"
TransactionIdentifier="68473921" TimeStamp="2010-02-11T16:36:06" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Success/>
    <Authorization>
        <AuthorizationDetail>
            <CreditCardAuthorization TransactionType="Sale" SaleCode="No Show Charge"
AuthorizationCode="4567" Amount="467.36">
                <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" CardCode="VI"
ExpireDate="0212"/>
                <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>
            </CreditCardAuthorization>
            <BookingReferenceID Type="14" ID_Context="PMSABC" ID="567841" />
        </AuthorizationDetail>
        <AuthorizationResult ApprovalDateTime="2010-02-11T16:36:03" AddressResultCode="Z"
AuthorizationCode="4567" Result="Approved" Description="Sale successful"/>
    </Authorization>
</HTNG_PaymentCardProcessingRS>
```

6.2.6 Void Settlement (or Sale)

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRQ EchoToken="82c208c0-128e-475a-8423-ed06951898f5"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:47" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <POS>
        <Source TerminalID="192.168.12.82"/>
    </POS>
    <AuthorizationDetail>
        <CreditCardAuthorization TransactionType="Void Settlement" SaleCode="Normal Charge"
Amount="526.28" SourceType="NormalTransaction">
            <CreditCard CardNumber="412345678901234" ExpireDate="0212"/>
            <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>
        </CreditCardAuthorization>
    </AuthorizationDetail>
    <StayInfo>
        <HotelReservation>
            <RoomStays>
                <RoomStay>
                    <TimeSpan End="2010-03-01" Start="2010-03-04"/>
                </RoomStay>
            </RoomStays>
        </HotelReservation>
    </StayInfo>
</HTNG_PaymentCardProcessingRQ>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRS EchoToken="82c208c0-128e-475a-8423-ed06951898f5"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:53" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Success/>
    <Authorization>
        <AuthorizationDetail>
            <CreditCardAuthorization TransactionType="Void Settlement" SaleCode="Normal
Charge" AuthorizationCode="4146" Amount="526.28">
                <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" CardCode="VI"
ExpireDate="0212"/>
                <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>
            </CreditCardAuthorization>
            <BookingReferenceID Type="14" ID_Context="PMSABC" ID="567841" />
        </AuthorizationDetail>
```

```
<AuthorizationResult ApprovalDateTime="2010-02-12T01:21:05"
AuthorizationCode="4146" Result="Approved" Description="Void Settlement successful"/>
</Authorization>
</HTNG_PaymentCardProcessingRS>
```

6.2.7 Return

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRQ EchoToken="def9ea42-4cc3-40b7-bdc7-641b78304db3"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:37" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <POS>
    <Source TerminalID="192.168.12.80"/>
  </POS>
  <AuthorizationDetail>
    <CreditCardAuthorization TransactionType="Return" SaleCode="Normal Charge"
Amount="526.28" SourceType="NormalTransaction">
      <CreditCard CardNumber="412345678901234" ExpireDate="0212"/>
      <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>
    </CreditCardAuthorization>
  </AuthorizationDetail>
  <StayInfo>
    <HotelReservation>
      <RoomStays>
        <RoomStay>
          <TimeSpan End="2010-03-01" Start="2010-03-04"/>
        </RoomStay>
      </RoomStays>
    </HotelReservation>
  </StayInfo>
</HTNG_PaymentCardProcessingRQ>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_PaymentCardProcessingRS EchoToken="def9ea42-4cc3-40b7-bdc7-641b78304db3"
TransactionIdentifier="68473921" TimeStamp="2010-02-12T01:21:43" Version="1.0"
xmlns="http://htng.org/2010A" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Success/>
  <Authorization>
    <AuthorizationDetail>
      <CreditCardAuthorization TransactionType="Return" SaleCode="Normal Charge"
AuthorizationCode="1747" Amount="526.28">
        <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" CardCode="VI"
ExpireDate="0212"/>
        <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>
      </CreditCardAuthorization>
      <BookingReferenceID Type="14" ID_Context="PMSABC" ID="567841" />
    </AuthorizationDetail>
    <AuthorizationResult ApprovalDateTime="2010-02-12T01:21:55"
AuthorizationCode="1747" Result="Approved" Description="Return successful"/>
  </Authorization>
</HTNG_PaymentCardProcessingRS>
```

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

7.1 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. 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/>

7.2 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

7.3 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.

7.4 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)

7.5 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