Payment Systems & Data Security Specifications 2010B

Payment Processing Specification Version 2.0
22 October 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.

Table of Contents

3   Document History

3.1   Functional Change Log

Version	Date	Comments
1.0	23 Apr 2010	Support for lodging transactions: • Authorization • Incremental Authorization • Authorization Reversal • Settlement • Sale (No Prior Authorization) • Void Settlement • Return
2.0	22 Oct 2010	Added support for additional verticals (Restaurant, Retail, eCommerce), Partial Authorizations and Authorizations with Balance Return (for use with Prepaid cards), and EMV.
2.0 Errata	18 Apr 2012	Updated error – Central Reservation System now refers to CRS, instead of CRM

4   Document Information

4.1   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.

New features added in the 2010B release:
- Support for additional verticals (Restaurant, Retail, eCommerce)
- Partial Authorizations and Authorizations with Balance Return (for use with Prepaid cards)
- EMV (Chip and PIN)

4.2   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.

4.3   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.

4.4   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.

4.5   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	**A**ddress **V**erification **S**ystem – 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	**B**ank **I**dentification **N**umber – ranges from 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. Magnetic Stripe Codes CVV – Card Verification Value (Visa and Discover) CVC – Card Validation Code (MasterCard) CAV - Card Authentication Value (JCB) CSC – Card Security Code (American Express) Visible Codes 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)
CRS	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 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 cannot offer Authorization to a transaction request for whatever reason (i.e. insufficient funds, stolen card, invalid card etc.).
EMEA	**E**urope, **M**iddle **E**ast and **A**frica – generic trading zone.
EMV	**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	**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	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. There are three magnetic stripe tracks on credit cards. 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. Track 2 can hold up to 40 characters. Typical track 2 data includes PAN, expiration date, and discretionary data. 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.
Merchant	The organization or business providing goods or services to the cardholder.
MID	**M**erchant **Id**entifier **-** 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	**N**ear **F**ield **C**ommunications – 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	**P**rimary **A**ccount **N**umber 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	**P**IN **E**ntry **D**evice **–** a common name for a chip card reader that has a built in PIN pad.
PIN	**P**ersonal **I**dentification **N**umber – primarily used with debit and/or Chip card transactions
PMS	**P**roperty **M**anagement **S**ystem – 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	**P**oint **O**f **S**ale – 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	**R**adio **F**requency **Id**entification – 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 that 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	**T**erminal **Id**entifier **–** 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	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.

4.6   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 Alliance Specifications	http://www.opentravel.org
PCI-DSS documentation	https://www.pcisecuritystandards.org/

5   Business Process

5.1   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, Restaurant, Retail and eCommerce 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.

5.2   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.

5.2.1   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)
- Restaurant Point of Sale Systems
- Retail Point of Sale Systems
- eCommerce Systems

5.2.2   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.

5.3   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).

5.3.1   Reservation Process

5.3.1.1 Reservation Made with Guest Providing Card Number – No Authorization Is Secured
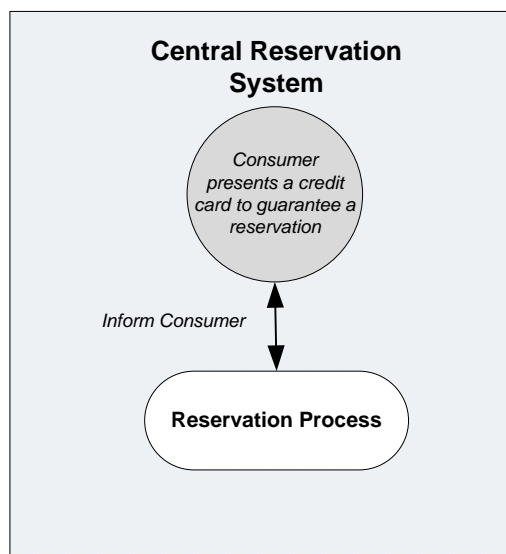
5.3.1.2 Reservation Made with Guest Providing Card Number – Payment Is Secured with Card and Transaction Is Settled
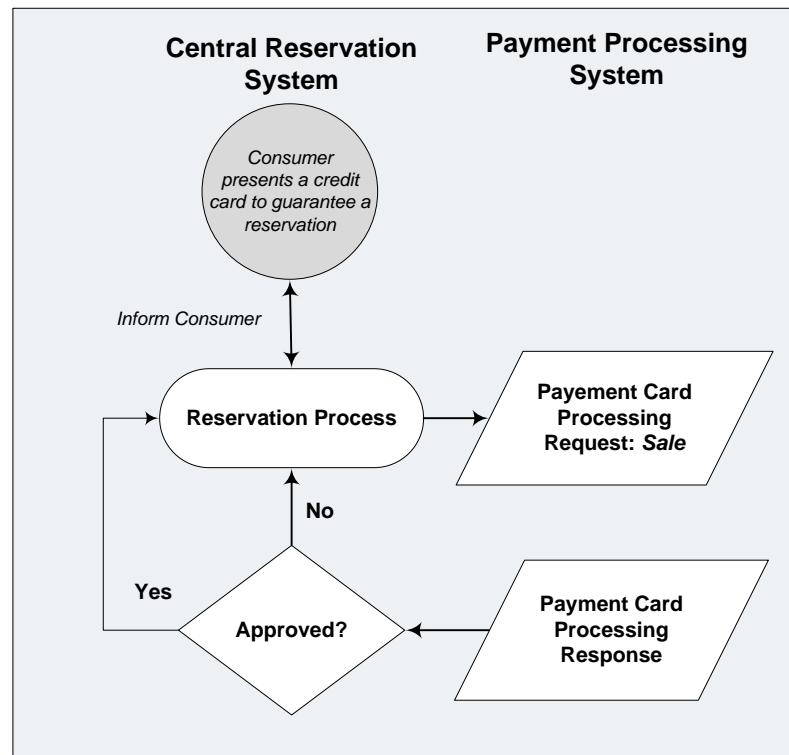
5.3.2   Change Card Process

5.3.2.1 Guest Changes Card Being Used – Change Takes Place During Stay
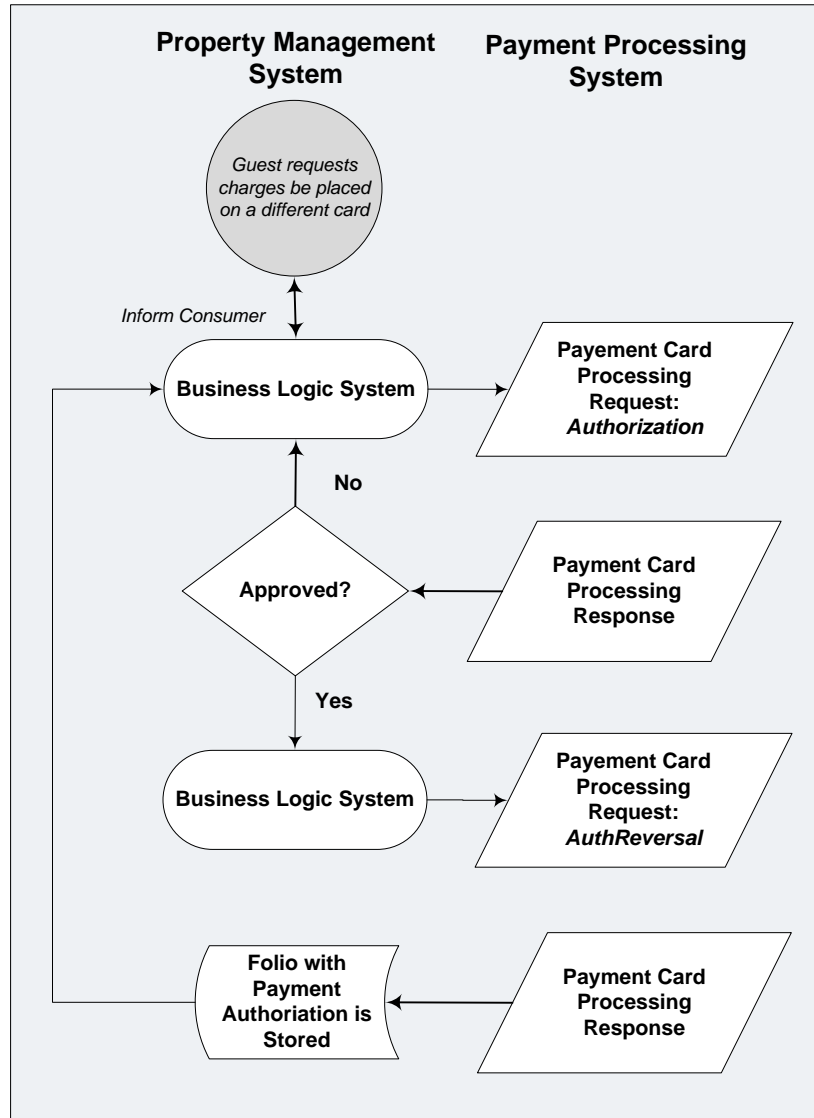
5.3.3   Guest Check-In Process

5.3.3.1 Guest Checks In – Same Card Used for Reservation Is Presented
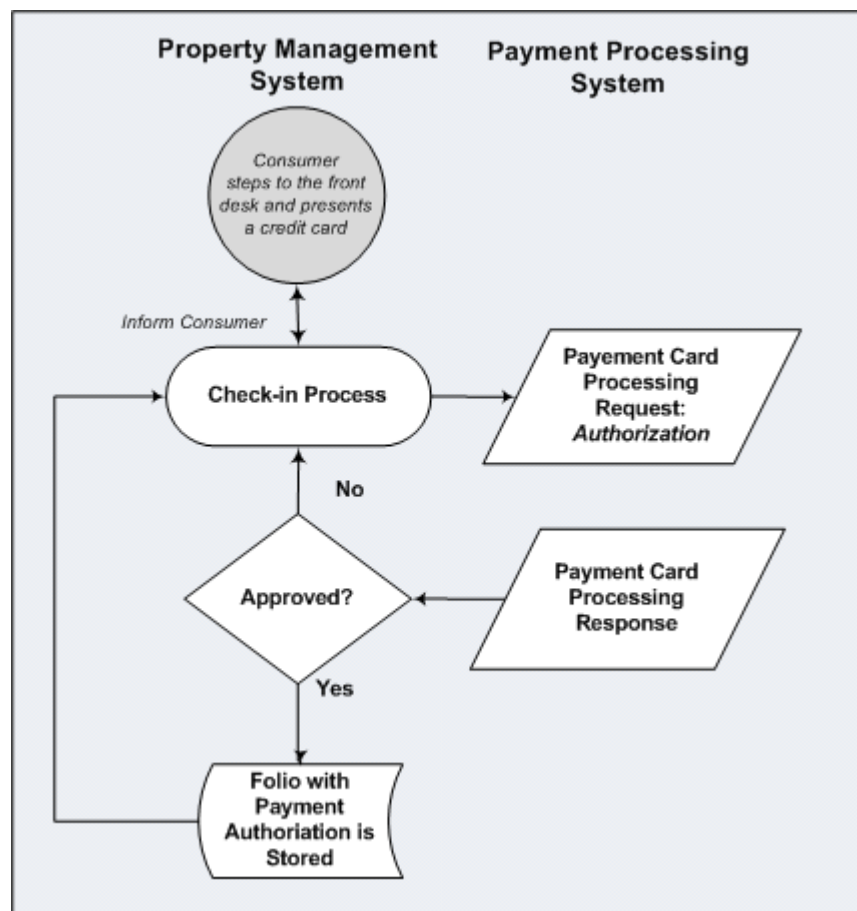
5.3.3.2 Guest Checks In with No Card – Uses Card-On-File Instead

Use case diagram is the same as above.

5.3.4   During Stay Activity

5.3.4.1 Checked-In Guest Accumulates More Charges than Previously Authorized – Business Logic System Recognizes Situation and Obtains an Incremental Authorization
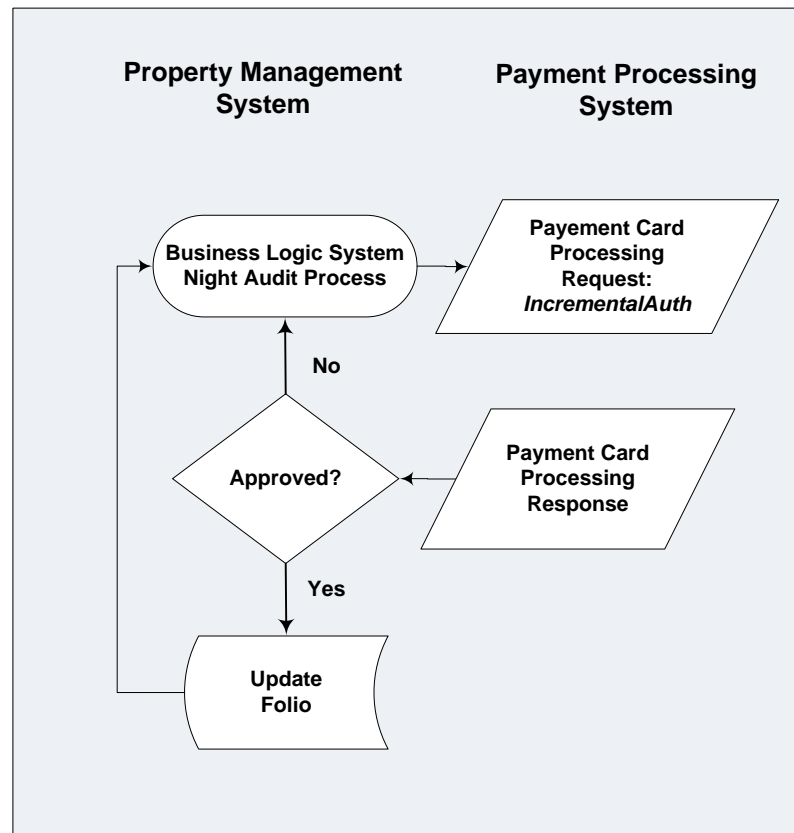
5.3.5   Guest Check-Out Process

5.3.5.1 Guest Is Checking Out – Accumulated Folio Charges Exceed Amount Currently Authorized on Guest's Card
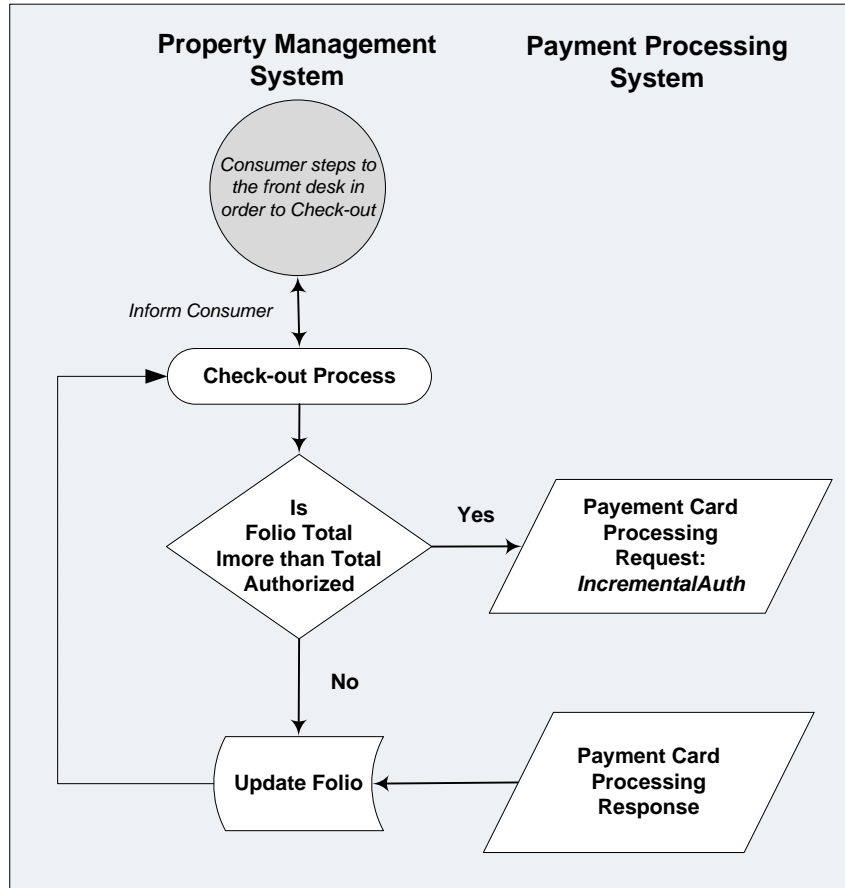
5.3.5.2 Guest Is Checking Out – Card Has Higher Amount Authorized than Accumulated in the Folio
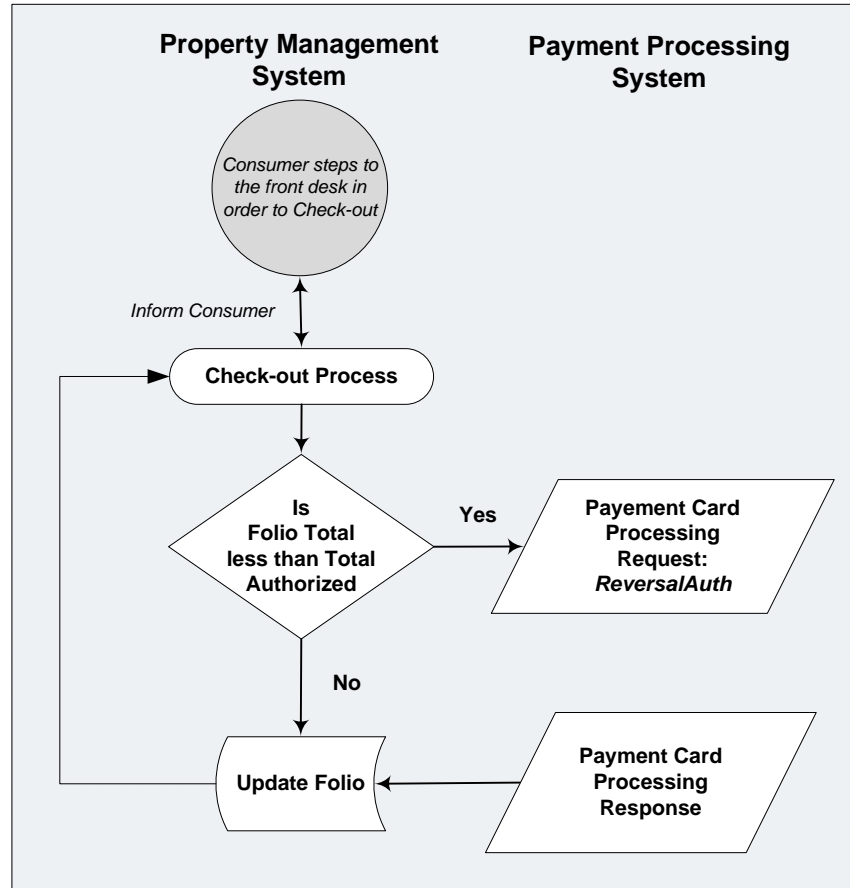
NOTE: Use case presumes that Folio Total cannot exceed Total Authorized as any attempt to accumulate a folio charge greater than the approved amount would result in an incremental authorization.

5.3.5.3 Upon Completing Transaction Adjustments at Check-Out – Business Logic System Marks Transaction as "Settleable" for Funds Capture with Next Batch Close Process
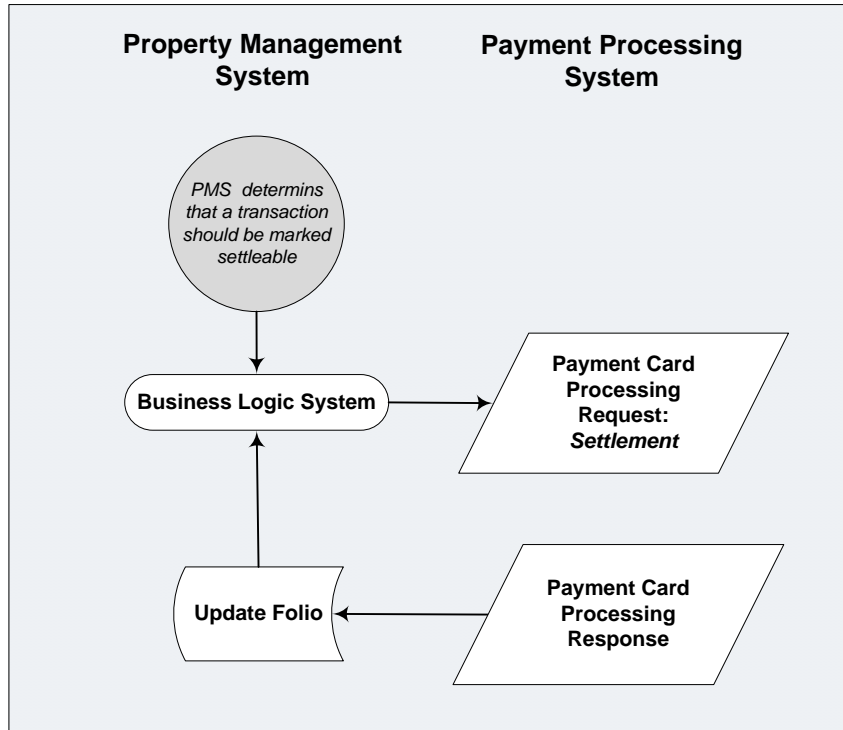
5.3.6   Extended Stay Processing

5.3.6.1 Guest Extends Stay – Business Logic System Marks Guest's Current Transaction as "Settleable" for Funds Capture
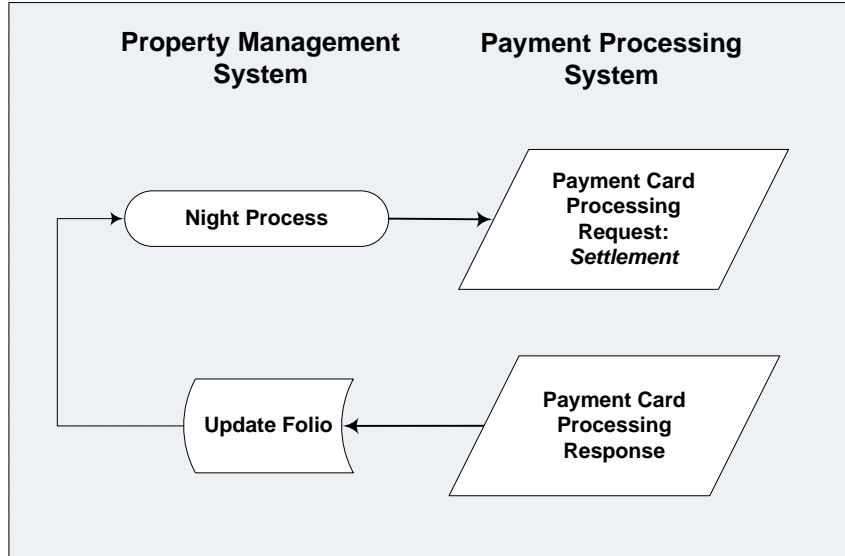
5.3.7   Cancellation/No-Show

Cancellation/No-Show use cases are the same as Check-Out – no additional use cases are needed. Refer to 5.3.5

5.3.8   Batch Close

Currently, this specification does not support a batch close process. Refer to 6.8.1

5.3.9   Void Settlement

5.3.9.1 Marking a Transaction as Not "Settleable"

Since the payment also exists in the Payment Processing System, the Business Logic System should invoke a Void Settlement Use Case to suspend or cancel a previous request to settle a transaction. Refer to 6.9
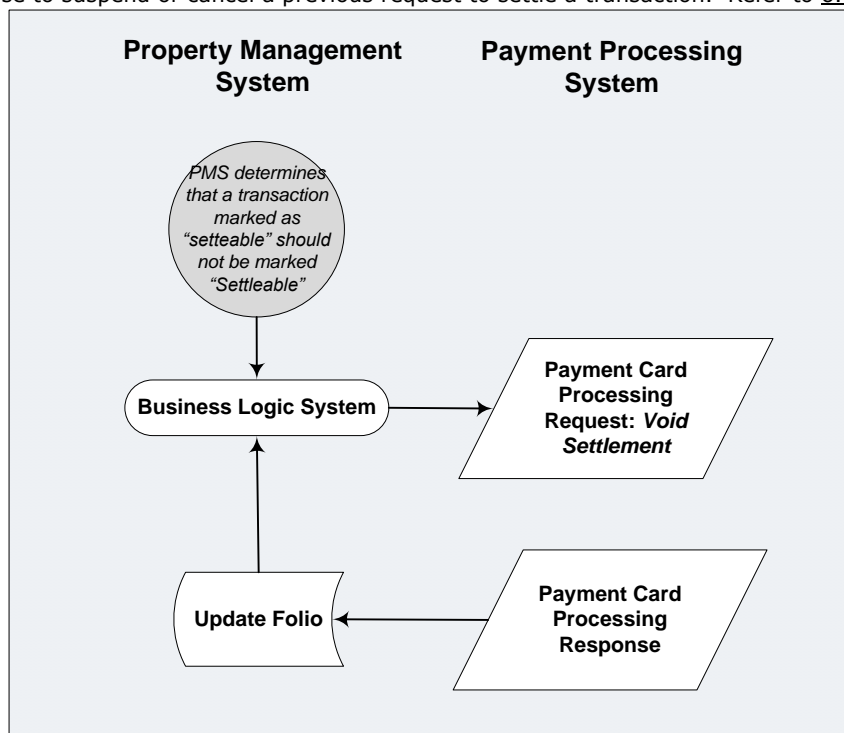
5.3.10 Return Transaction

5.3.10.1
Required

Property Management System or Hotel Personnel Determine(s) a Return Transaction is
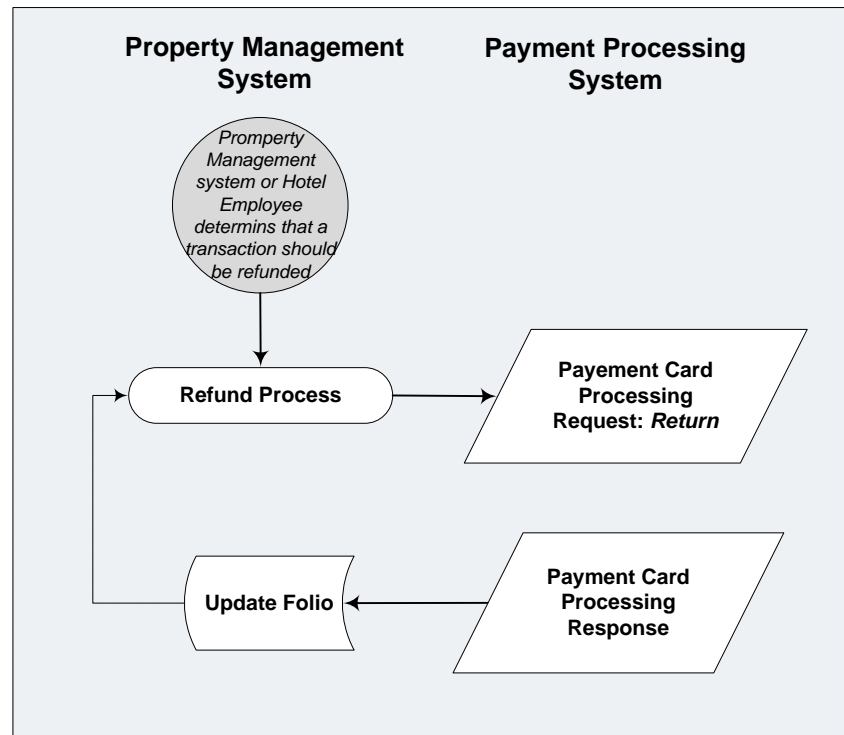
5.3.11 Post Departure Charge

5.3.11.1   **Guest Has Checked Out – Charge Needs to be Applied to Guest's Card**
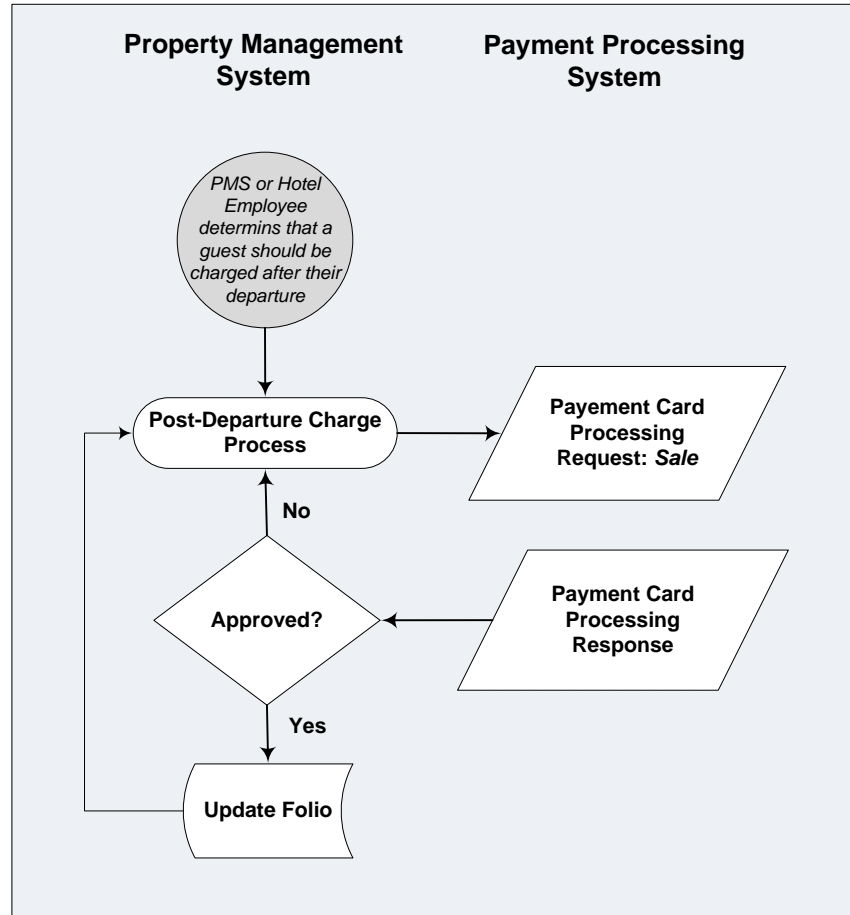
5.3.12   Sale Transaction for Card-Present Retail

5.3.12.1   Used in Gift Shops or Other Retail Outlets in Hotel

Refer to 6.12
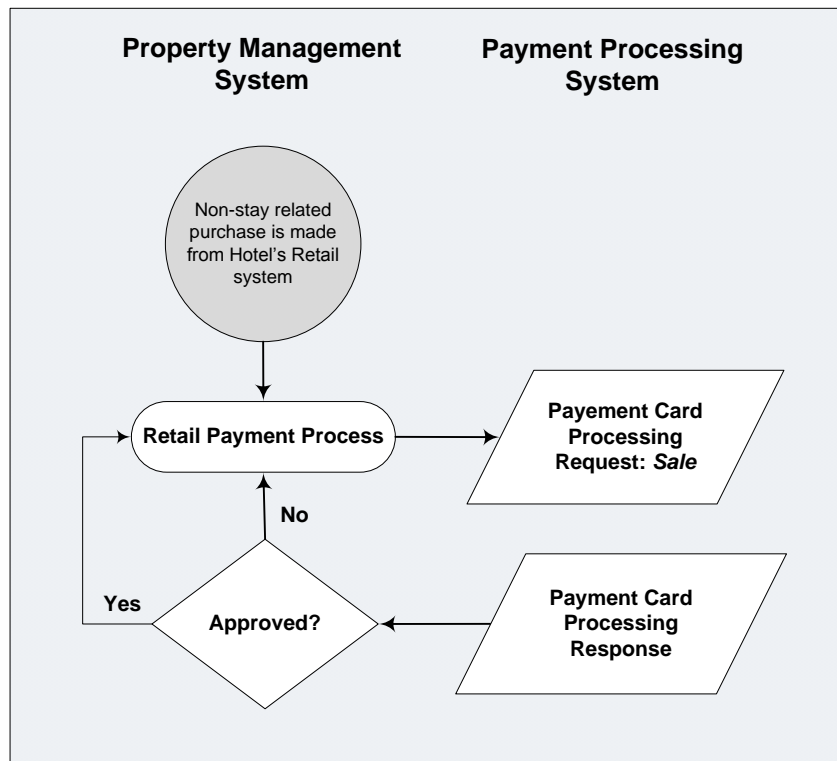
5.3.13 Pick and Ship

5.3.13.1   Authorization for Shipping

An item is ordered from an on-line site. The card is authorized for the estimated total amount of the sale, including tax and shipping charges. Refer to 6.12.7
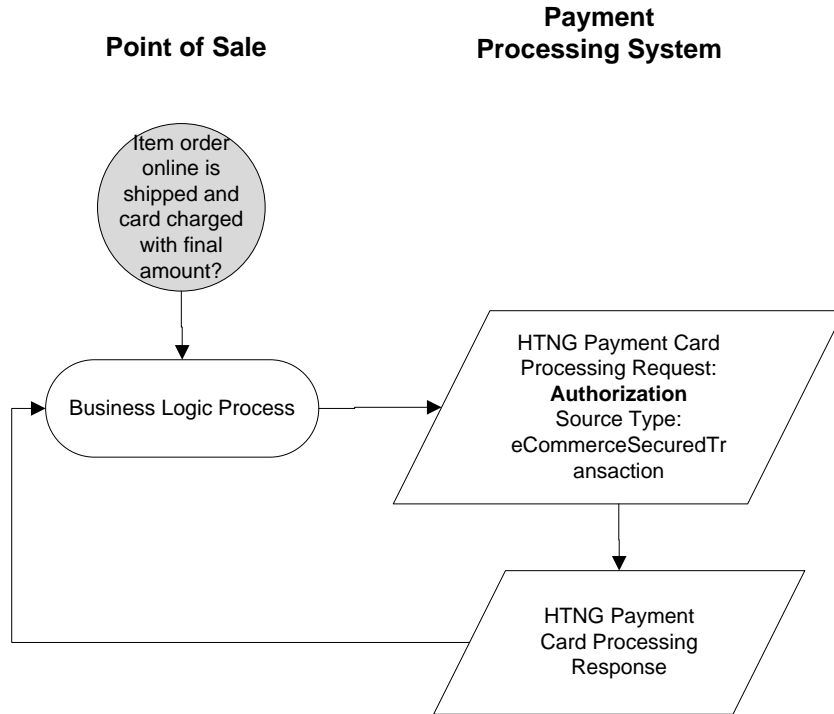
5.3.13.2   Settlement After Shipping

The item previously ordered from the on-line site ships. The card is charged the final amount. Refer to 6.12.8

Point of Sale

Payment Processing System

Item order online is shipped and card charged with final amount?

Business Logic Process

HTNG Payment Card Processing Request:
Settlement
Source Type: eCommerceSecuredTransaction

HTNG Payment Card Processing Response
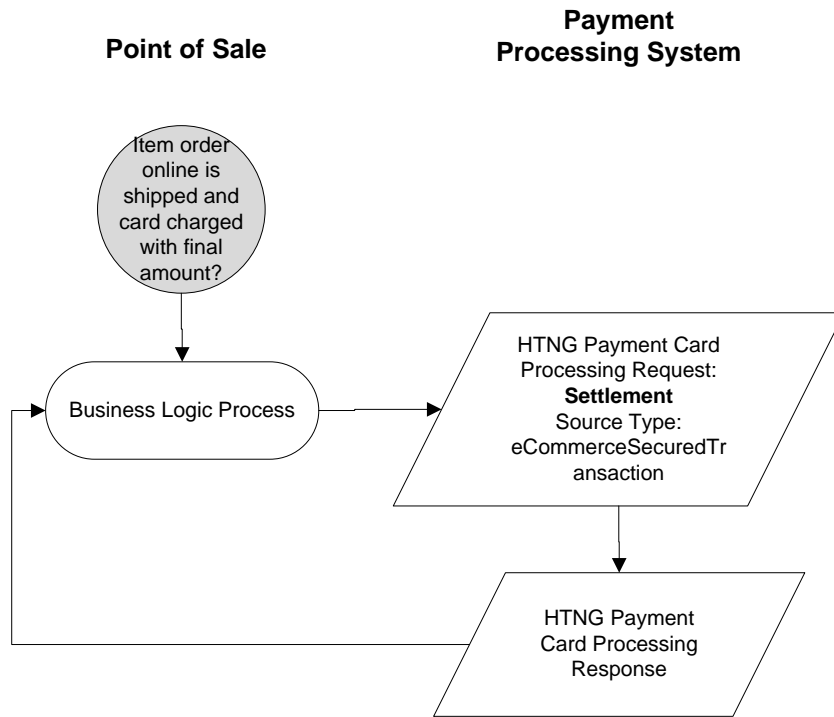
5.3.14 Settlement with Gratuity

5.3.14.1   Guest Leaves Tip on Credit Card Voucher

Refer to 6.12.1

Point of Sale

**Payment
Processing System**

```
        ( Guest makes a
          Payment at the
          Resturant in the
          Hotel )
               |
               v
    +-------------------+              /--------------------------/
    |                   |------------>/  HTNG Payment           /
    | Business Logic    |            /  Processing Request:    /
    | Process           |            /  Authorization          /
    |                   |            /------------------------/
    +-------------------+
               |
               v
        < Guest adds Tip to the >          /--------------------------/
        < initial amount paid   >--YES-->/  HTNG Payment            /
        <                       >       /  Processing Request      /
                                        /  with Final amount       /
                                        /  including Tip:          /
                                        /  Settlement              /
                                        /------------------------/
                                                  |
                                                  v
                                        /--------------------------/
                                       /  HTNG Payment            /
                                      /  Processing Response      /
                                     /------------------------/
```
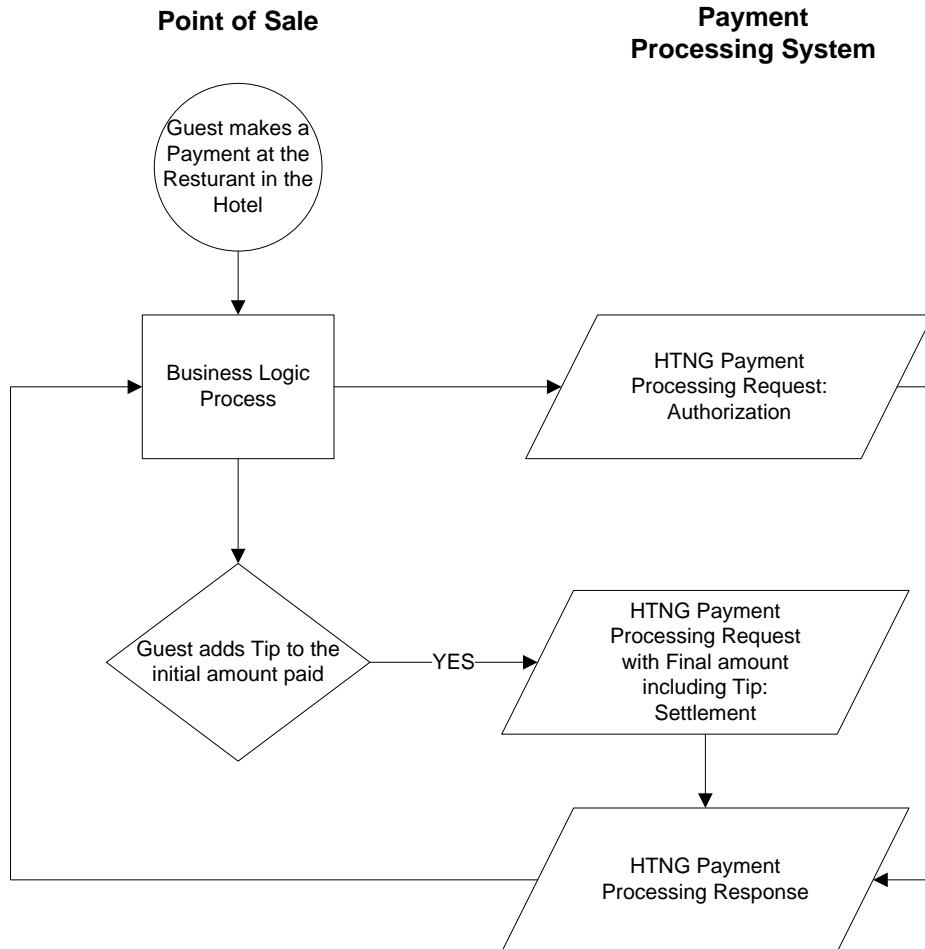
5.3.14.2   Guest Dines in Restaurant Where Tips Are Not Applicable

Refer to 6.12.2

Point of Sale

**Payment
Processing System**

Guest makes a
Payment with NO TIP
applicable

Business Logic
Process

HTNG Payment
Card Processing
Request: Sale

HTNG Payment
Card Processing
Response
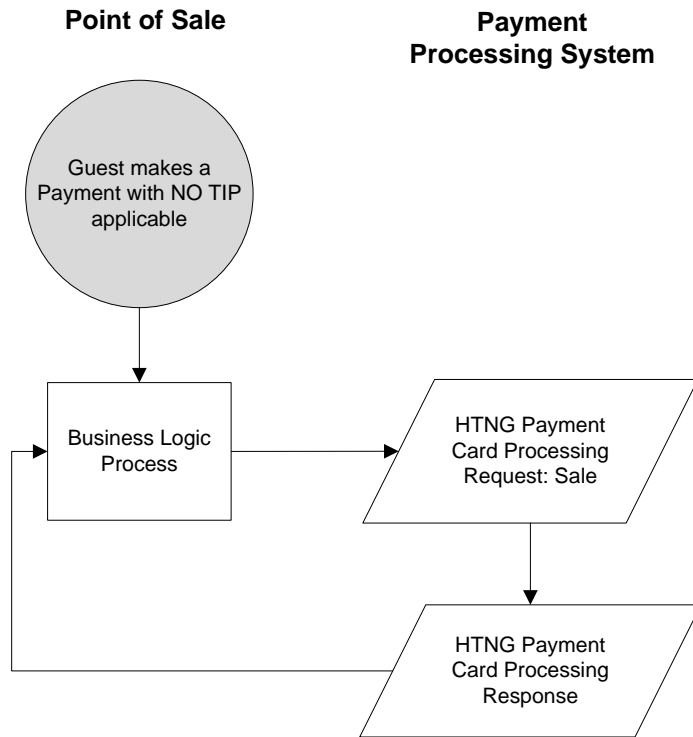
5.3.15 Prepaid Cards

5.3.15.1   **Payment Using Prepaid Card**

Transaction authorized with prepaid card – if balance available is less than authorization attempt, card is authorized for partial amount. If the available balance on the card is known, it is returned. Refer to 6.12.4

Point of Sale   **Payment Processing System**
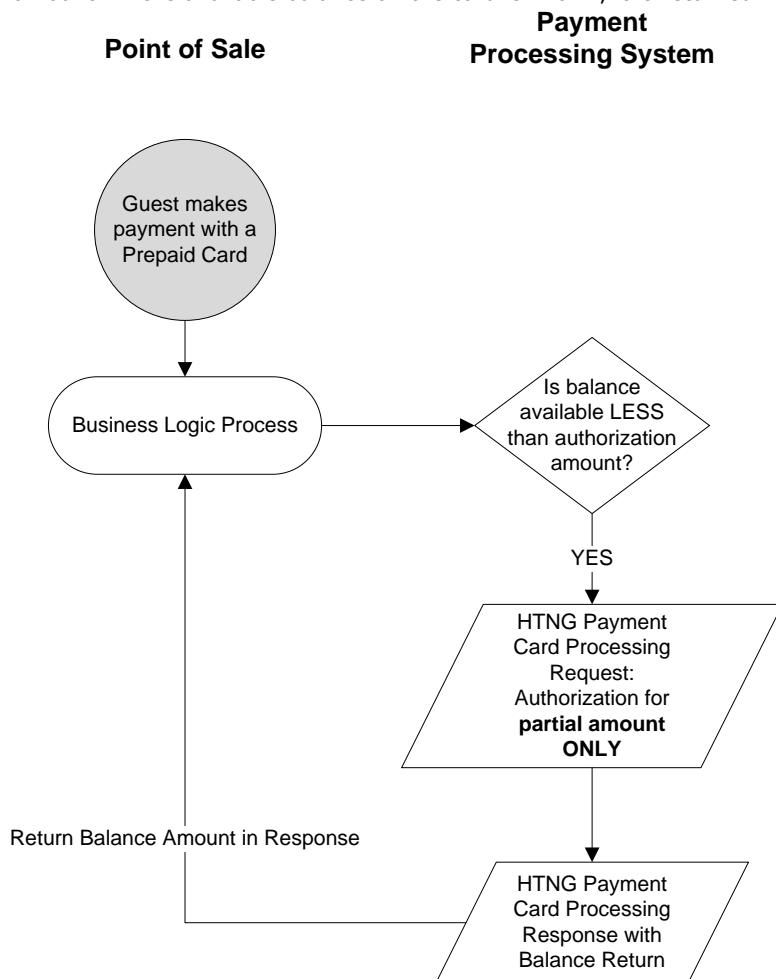
5.3.15.2   Guest Asks for Balance of Prepaid Card

Refer to 6.12.5

Point of Sale   **Payment Processing
                                          System**
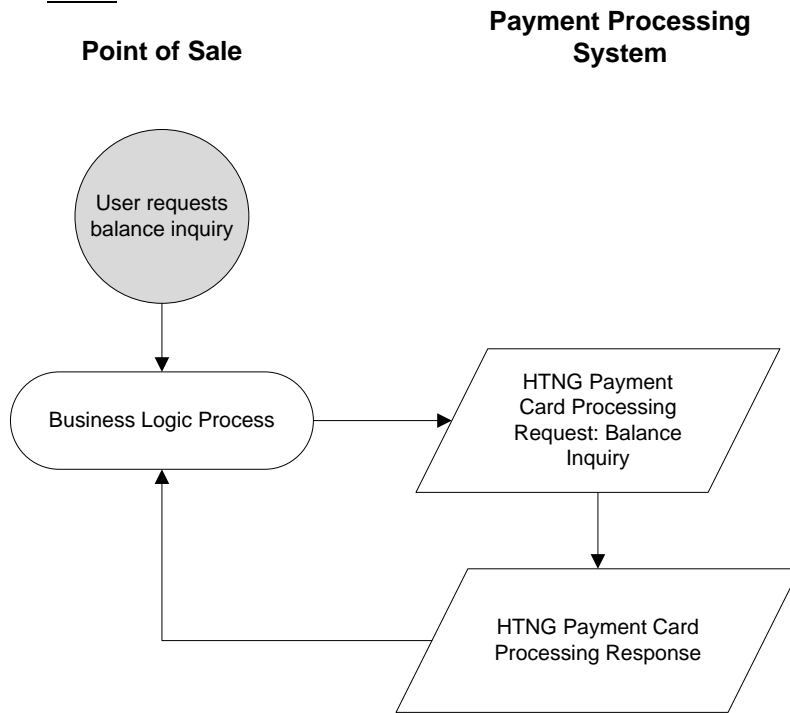
5.3.16 Authorization Reversal

5.3.16.1   Transaction Authorized – Then Removed Before Finalized (Before Settled)

Refer to 6.12.3

Point of Sale

**Payment
Processing System**

Guest makes a
payment

Business Logic
Process

HTNG Payment
Card
Processing
Request:
Authorization

Has amount been
removed or reduced
before final settlement?

YES

HTNG Payment
Card
Processing
Request:
AuthReversal

HTNG Payment
Card Processing
Response
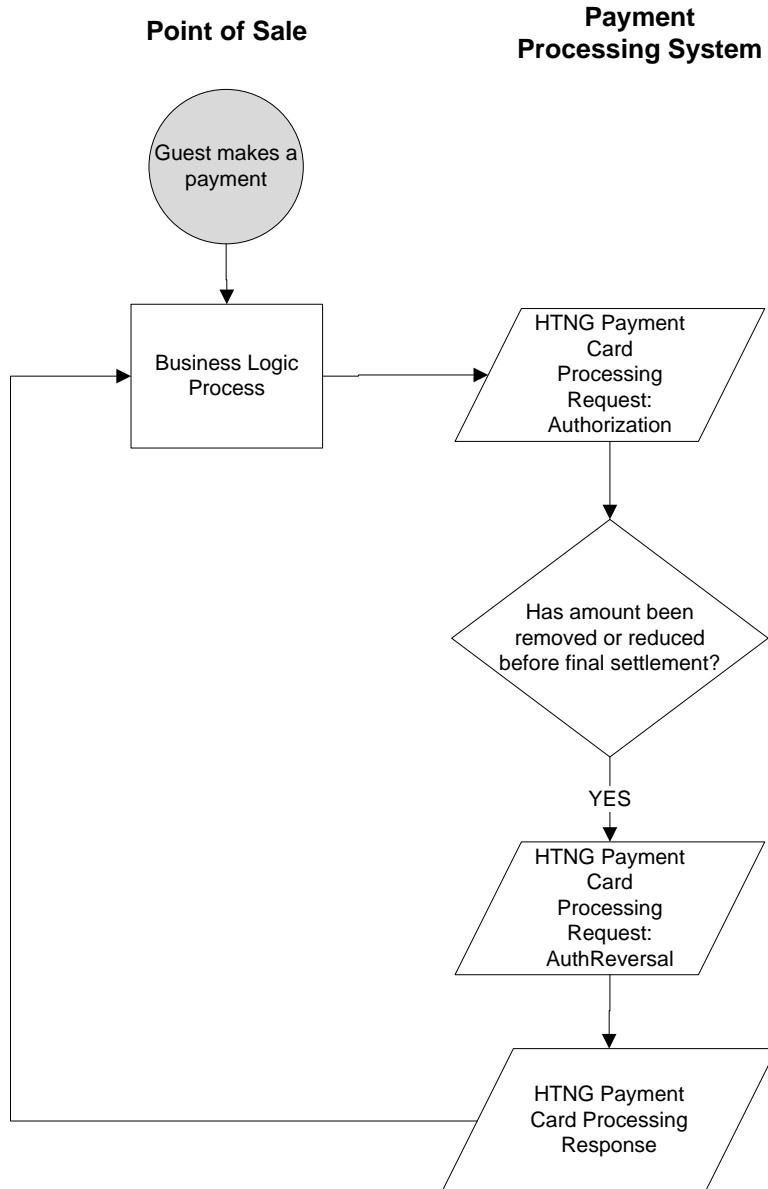
5.3.17 Voice Authorization

5.3.17.1   Actor Is Unable to Perform an On-Line Authorization with Payment Processing System

In this case, the Actor will perform a Voice Authorization, where no transaction is sent to the Payment Processing System. Refer to 6.12.6

Point of Sale
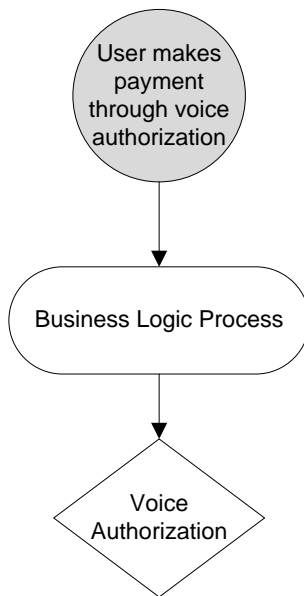
5.3.18 EMV Flow Diagrams

5.3.18.1   Check-In With Fixed Device

Guest checks into the hotel. When prompted, cardholder inserts card into the PED (PIN Entry Device). The PED displays an "Estimated Amount" and prompts the cardholder for their PIN. The cardholder keys in PIN and the transaction is authorized and the cardholder is prompted to remove their card. Refer to 6.12.10 and 6.12.11

Check-in with Fixed PED device
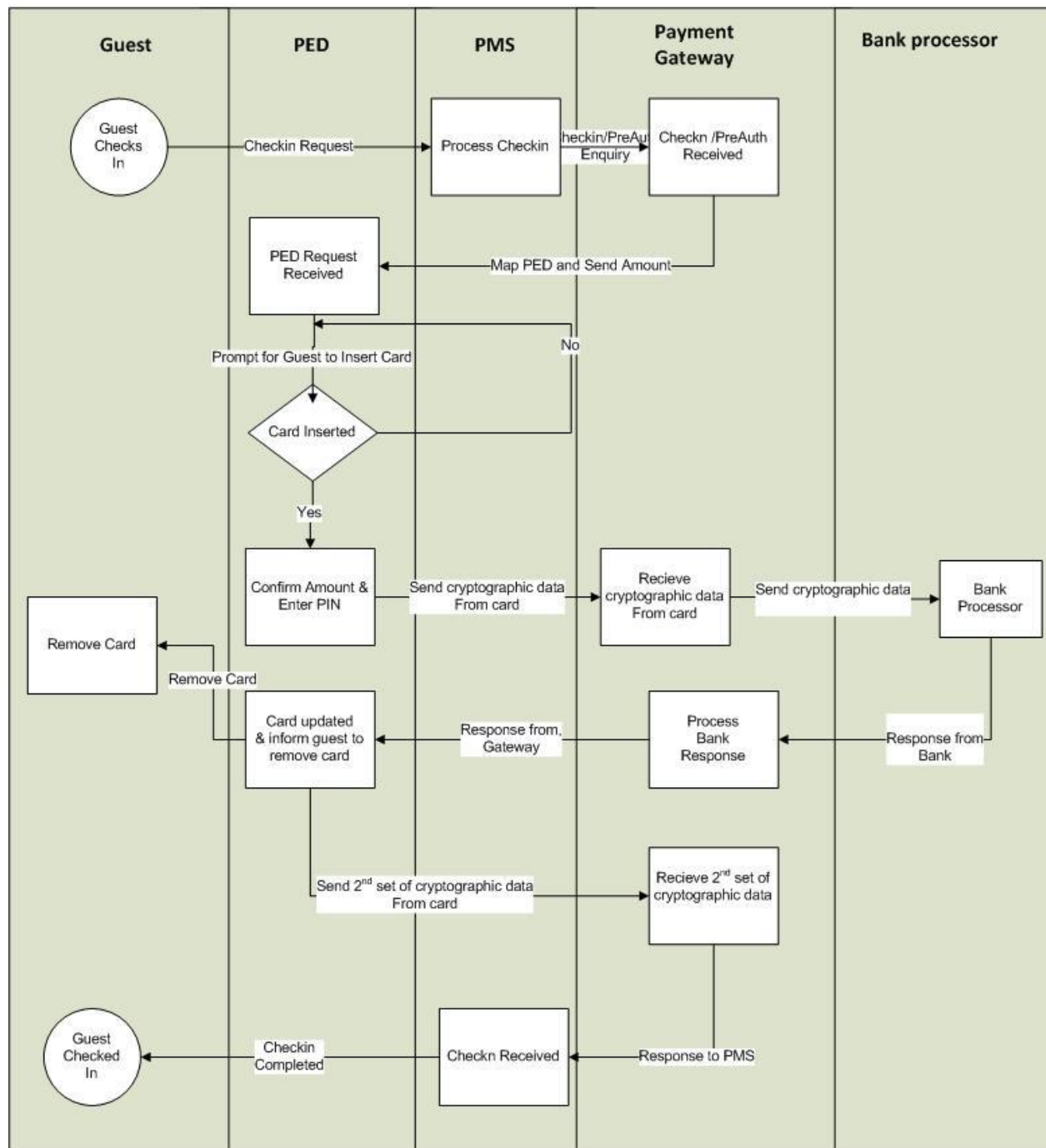
5.3.18.2   Guest Checks In With Same or Different Card

Guest checks into the hotel. When prompted, cardholder inserts card into the PED. The PED displays an "Estimated Amount" and prints receipt. Cardholder signs receipt. Refer to 6.12.10 and 6.12.11

Check-in with Same or Different Card
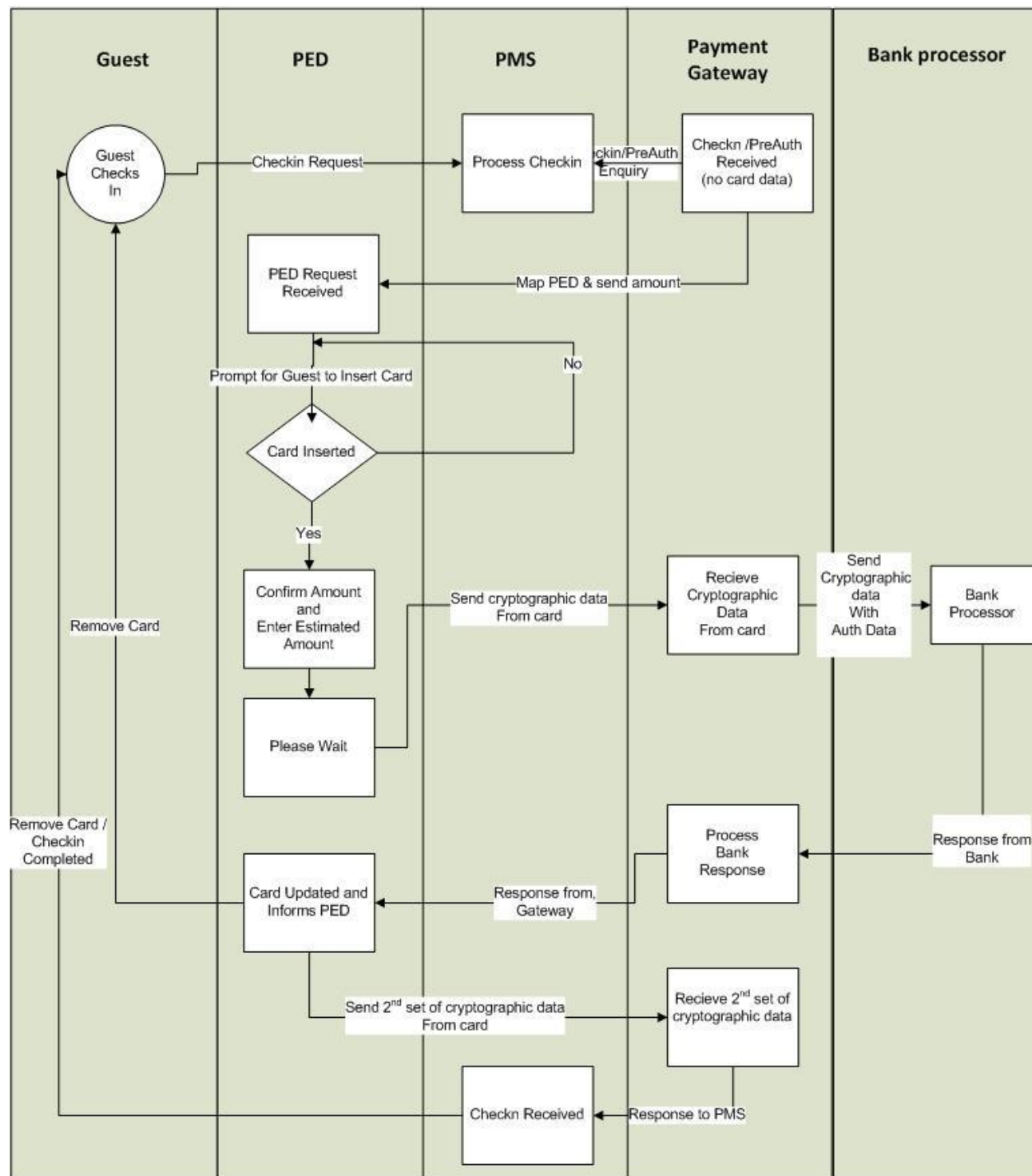
5.3.18.3 Fine Dining With Wireless PED

At conclusion of meal, a diner calls for their check. Waiter finalizes the itemized bill and closes the table in the POS. Waiter takes bill and PED to the table. Diner accepts the bill amount displayed on the PED, enters gratuity, inserts card and enters PIN. After approving transaction, diner is prompted to remove their card. Refer to 6.12.12

Fine Dining with Wireless PED
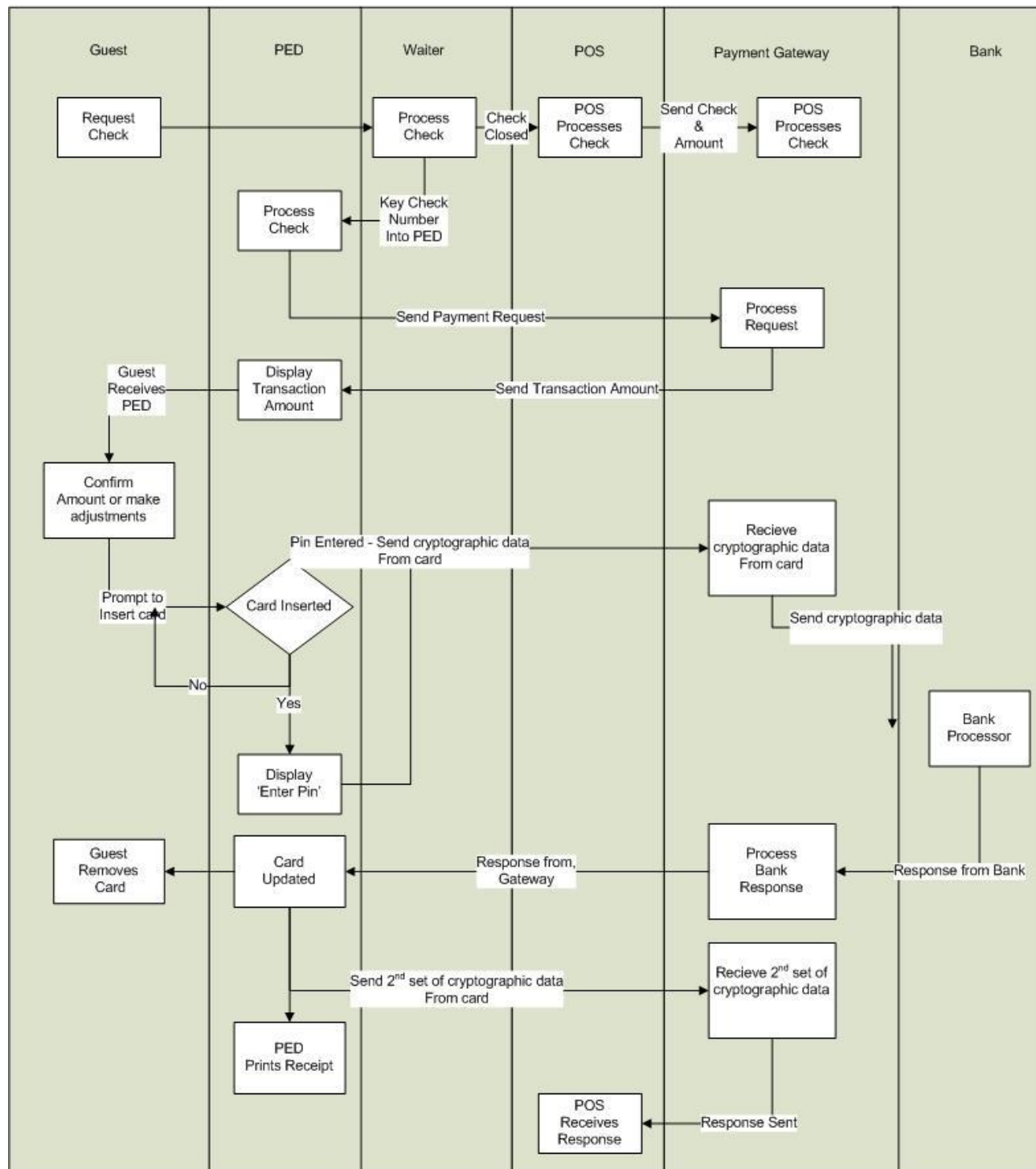
5.3.18.4   Dining With Fixed PED at POS

Waiter leaves check at table. When ready, diner takes check to cashier and cashier closes the check. Check total is displayed on local PED. Cardholder adjusts the total with gratuity, inserts card and enters PIN. Upon approval, cardholder is requested to remove card. Refer to 6.12.13

Dining with Stationary PED at POS
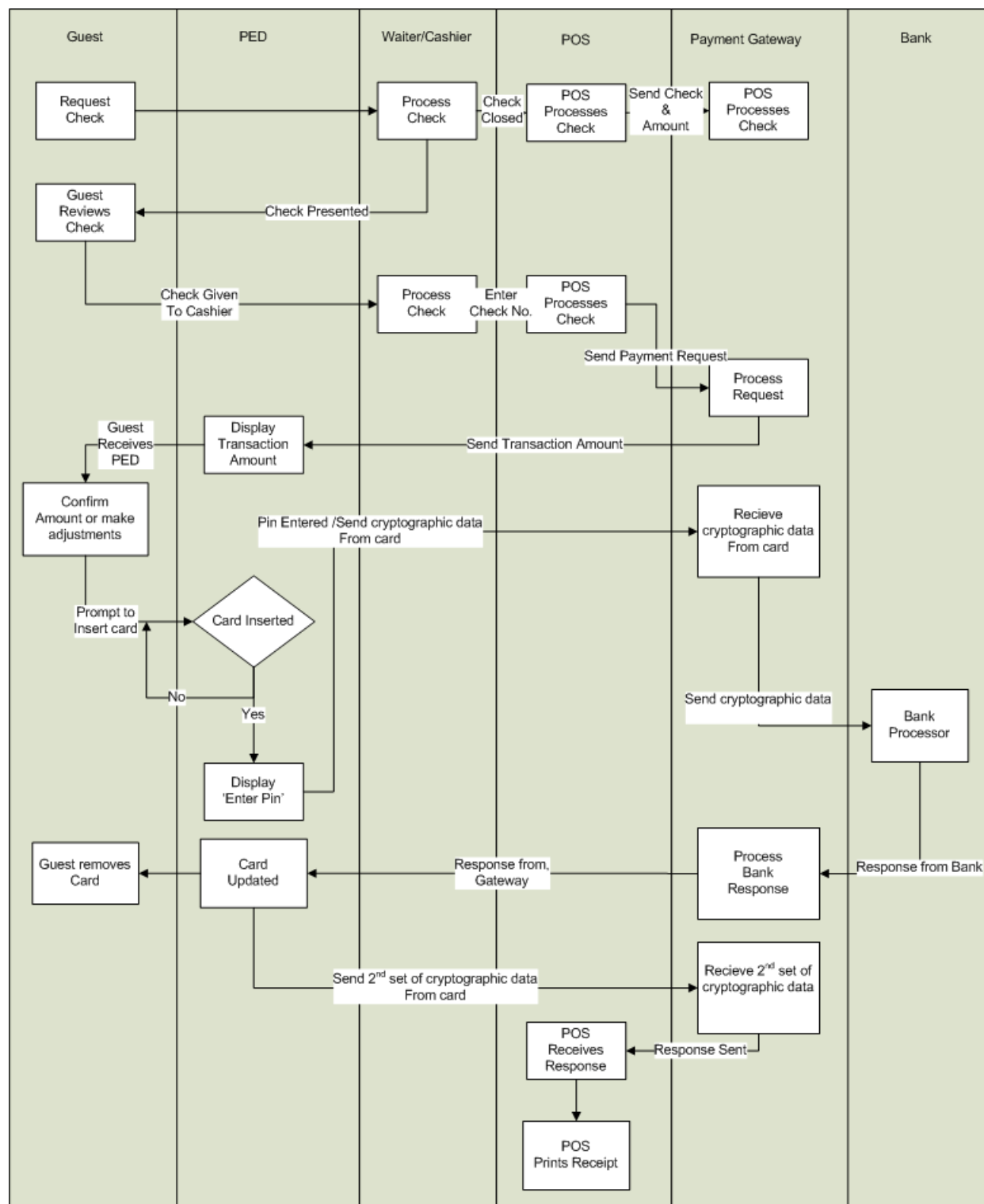
5.3.18.5   Check-Out

A guest has checked out. The transaction is sent to the Payment Processing System that includes crypto data for later settlement. Refer to 6.5.3
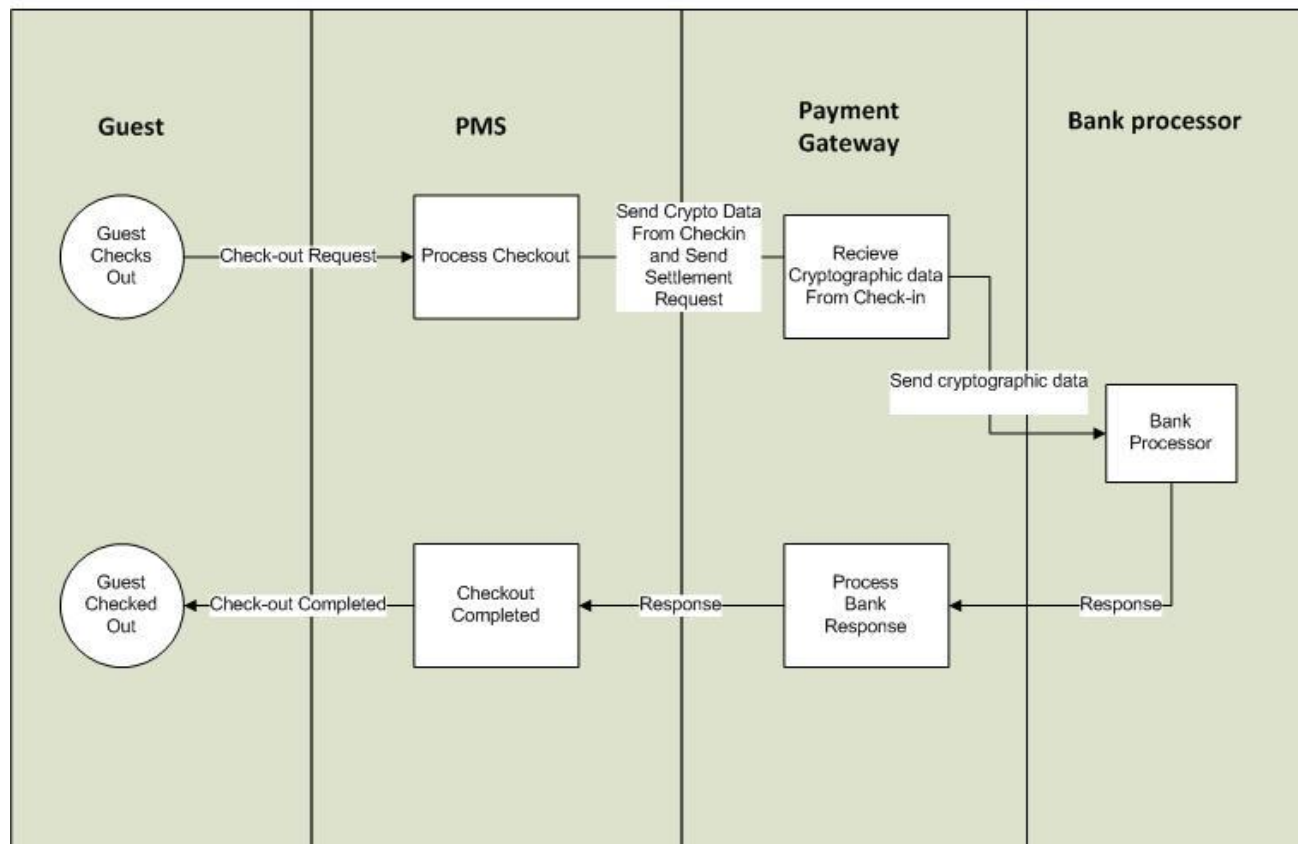
5.3.18.6   Extended Stay Settlement

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. Refer to 6.6

Extended Stay Settlement
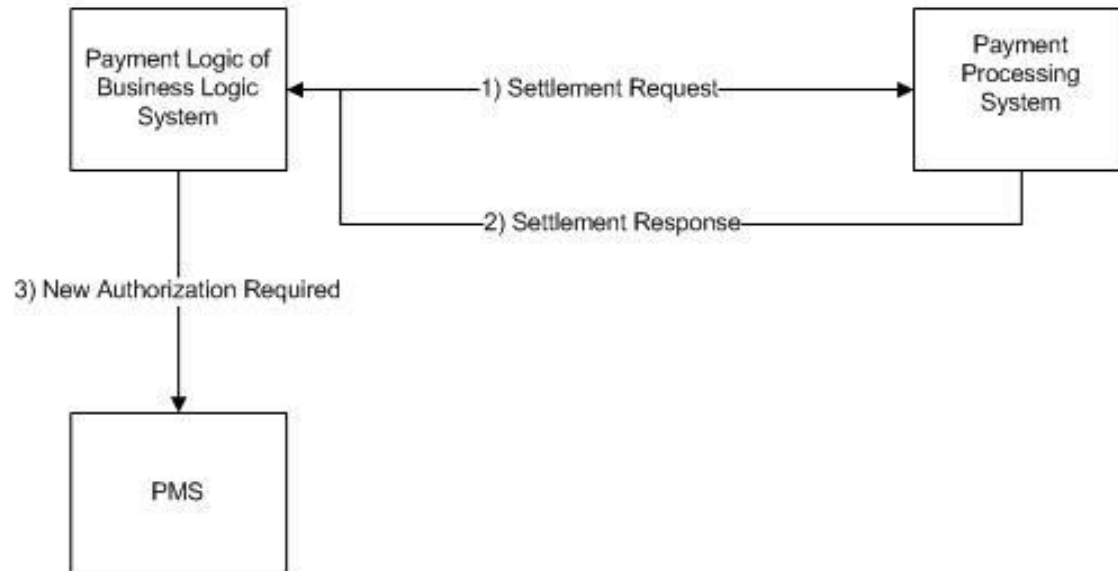
6   Use Cases

6.1   Reservations

6.1.1   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.

6.1.2   Reservation, Hold Card Number, 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.

6.2   Change Card

6.2.1   Change Card That Is Stored With 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.

6.3   Check-In

6.3.1   Check-In With 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.

6.3.2   Check-In With No Card

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.

6.4   Duration of Stay

6.4.1   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.

6.5   Check-Out

6.5.1   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.

6.5.2   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.

6.5.3   *Check-Out - Transaction 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.

6.6   Extended Stay Settlement

6.6.1   *Transaction 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.

6.7   Cancellation/No-Show

6.7.1   *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.

6.8 Batch Close

6.8.1 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.

6.9 Void Settlement

6.9.1 Void (Will Remove Transaction from 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

6.10 Return

6.10.1 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.

6.11 Post-Departure Charge

6.11.1 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.

6.12   Restaurant, Retail, eCommerce and EMV

6.12.1 Add Tip (Final Amount Different Than Initial Authorization)

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- Guest eats in a restaurant. Check is presented to guest. Guest pays with credit card (HTNG_PaymentCardProcessingRQ is sent with the TransactionType set to Authorization). Guest leaves a tip on the credit card voucher.

Basic Flow
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Settlement) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return. The Amount attribute will be the final amount, including the tip.
- The use case terminates.

Preconditions
- None

Postconditions
- The transaction is marked as Settled in the POS and "Settleable" in the Payment Processing System.

6.12.2 Sale (No Tip Transaction)

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- Guest eats in a restaurant where tips are not applicable.

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.
- The use case terminates.

Preconditions
- None

Postconditions
- None

6.12.3 Auth Reversal

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- The transaction was authorized, but then removed before it was finalized (before it was settled).
  Or
- The transaction was authorized for an amount, but then it will be finalized for a lesser amount.

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

Preconditions
- None

Postconditions
- None

6.12.4 Auth on Prepaid Card (Partial Auth with Balance Return)

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- The transaction is authorized with a Prepaid card. If the balance available on the card is less than the authorization attempt, the card is authorized for the partial amount. If available, the card balance is returned.

Basic Flow
- 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.
- If the balance available on the card is less than the authorization attempt, the card is authorized for the partial amount and the card balance is returned.
- The use case terminates.

Preconditions
- None

Postconditions
- If the balance available on the card is less than the authorization attempt, the card is authorized for the partial amount and the card balance is returned.
- The guest must present payment for the remaining amount.

6.12.5 Balance Inquiry

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- The guest presents a Prepaid card. They ask for the balance available on the card.

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

Preconditions
- None

Postconditions
- None

6.12.6 Voice Authorization

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- For whatever reason, the Actor is not able to perform an on-line authorization with the Payment Processing System. In that case, the Actor will perform a Voice Authorization, where no transaction is sent to the Payment Processing System. However, when the transaction is settled, it needs to be identified as having been Voice-Auth'd.

Basic Flow
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Settlement) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- We will add an "Authorized on-line yes/no" attribute to the Auth Request.
- The use case terminates.

Preconditions
- None

Postconditions
- None

6.12.7 Authorization Before Shipping (PreAuth)

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description

- An item is ordered from an on-line site. The card is authorized for the estimated total amount of the sale, including tax and shipping charges

Basic Flow
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Authorization and the SourceType set to eCommerceSecuredTransaction or eCommerceUnsecuredTransaction) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- The use case terminates.

Preconditions
- None

Postconditions
- None

6.12.8 Ship with Final Amount (PostAuth)

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- The item previously ordered from the on-line site ships. The card is charged the final amount.

Basic Flow
- The Actor issues an HTNG_PaymentCardProcessingRQ (with the TransactionType set to Settlement and the SourceType set to eCommerceSecuredTransaction or eCommerceUnsecuredTransaction) to the Payment Processing System and will receive an HTNG_PaymentCardProcessingRS in return.
- The use case terminates.

Preconditions
- None

Postconditions
- None

6.12.9 Europay, MasterCard and Visa (EMV)

Note – all of the below are the same for EMV transactions as for Card Swipe transactions, thus no additional use cases have been defined:
- Reservation, Hold card number, No Authorization
- Reservation, Hold card number, with Advance Deposit
- Incremental authorization
- Reversal authorization
- Check-out - transaction gets marked as "settleable"
- Extended Stay Settlement - Transaction gets marked as "settleable"
- Cancellation/No-show
- Batch Close
- Void
- Return
- Post-Departure Charge
- eCommerce (Sale) Transactions

6.12.10 Check-In (Authorization) From Fixed PED – Card Is Chip and PIN Enabled

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- Guest checks into the hotel. When prompted, cardholder inserts card into the PED (PIN Entry Device). The PED displays an "Estimated Amount" and prompts the cardholder for their PIN. The cardholder keys in PIN. The transaction is authorized and the cardholder is prompted to remove their card.

Basic Flow
- PMS sends check-in transaction to Payment Gateway (no card data)
- Payment Gateway maps nearest PED
- Payment Gateway drives PED and prompts cardholder to Insert or Swipe their card
- Payment Gateway receives initial card data and sends payment details
- PED displays 'Estimated Amount' and prompts Cardholder for PIN
- Cardholder keys in PIN

- PED displays 'Please Wait'
- PED sends Payment Gateway cryptographic data from the card
- Payment Gateway forwards card/cryptographic data to the Bank/Processor
- Payment Gateway receives response from Bank/Processor
- Payment Gateway forwards response to PED
- Card accepts response and informs PED
- PED returns 2nd set of cryptographic data to Payment Gateway
- PED Displays 'Remove Card'
- Payment Gateway responds to PMS
- PMS extracts card and cryptographic data
- Use case terminates

Preconditions
- None

Postconditions
- None

6.12.11 Check-In (Authorization) From Fixed PED – Card Is Chip and Signature Enabled

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- Guest checks into the hotel. When prompted, cardholder inserts card into the PED. The PED displays an "Estimated Amount". **NOTE:** Chip and signature is an intermediary step implemented by issuers as the market is maturing for chip and PIN.

Basic Flow
- PMS sends check-in transaction to Payment Gateway (no card data)
- Payment Gateway maps nearest PED (PIN Entry Device)
- Payment Gateway drives PED and prompts cardholder to Insert or Swipe their card
- Payment Gateway receives initial card data and sends payment details
- PED displays 'Estimated Amount' and prompts for Cardholder approval
- Cardholder accepts estimated amount
- PED displays 'Please Wait'
- PED sends Payment Gateway cryptographic data from the card
- Payment Gateway forwards card/cryptographic data to the Bank/Processor
- Payment Gateway receives response from Bank/Processor
- Payment Gateway forwards response to PED
- Card accepts response and informs PED
- PED returns 2nd set of cryptographic data to Payment Gateway
- PED Displays 'Remove Card'
- Payment Gateway responds to PMS
- PMS extracts card and cryptographic data
- Use case terminates

Preconditions
- None

Postconditions
- None

6.12.12 Restaurant Authorization from Wireless PED – With Tip

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- At conclusion of meal, a diner calls for their check. Waiter finalizes the itemized bill and closes the table in the POS. Waiter takes bill and PED to the table. Diner accepts the bill amount displayed on the PED, enters gratuity, inserts card and enters PIN. After approving transaction, diner is prompted to remove their card.

Basic Flow
- Waiter closes the check in the POS

- POS sends check number and amount to Payment Gateway
- Waiter takes bill and wireless PED to table (in practice, diner examines bill first)
- Waiter keys check number into PED
- PED sends Payment Request to Payment Gateway
- Payment Gateway sends PED transaction amount
- Waiter hands PED to the diner/cardholder
- PED displays amount and prompts cardholder to confirm/change amount (i.e., add gratuity or pay their portion of the bill)
- Cardholder makes desired amount adjustments (i.e., enters their portion of the bill and gratuity) and confirms amount
- PED prompts Cardholder to 'Insert or Swipe' their card
- Cardholder inserts card
- PED prompts cardholder to enter PIN
- Cardholder enters PIN. PED displays "Please Wait"
- PED sends Payment Gateway transaction amount and cryptographic data
- Payment Gateway sends PED authorization response from issuing bank
- Card accepts authorization response and informs PED
- PED sends second set of cryptographic data to Payment Gateway
- PED prompts cardholder to remove card
- PED prints transaction receipt
- Cardholder returns PED to waiter
- Payment Gateway sends response to POS
- Use case terminates

Preconditions
- None

Postconditions
- None

6.12.13 Restaurant Authorization from Fixed PED – No Tip

Provider	Business Logic System
Actor	Payment Logic of Business Logic System

Brief Description
- Waiter leaves check at table. When ready, diner takes check to cashier. Cashier closes the check. Check total is displayed on local PED. Cardholder adjusts the total with gratuity, inserts card and enters PIN. Upon approval, cardholder is requested to remove card.

Basic Flow
- Waiter closes the check in the POS
- Waiter takes check to table
- When ready, diner takes check to cashier
- Cashier enters check number into POS
- POS sends transaction amount to Payment Gateway
- Payment Gateway sends transaction details to nearest PED
- PED displays amount and prompts cardholder to confirm/change amount (i.e., add gratuity or pay their portion of the bill)
- Cardholder makes desired amount adjustments (i.e., enters their portion of the bill and gratuity) and confirms amount
- PED prompts Cardholder to 'Insert or Swipe' their card
- Cardholder inserts card
- PED prompts cardholder to enter PIN
- Cardholder enters PIN. PED displays "Please Wait"
- PED sends Payment Gateway transaction amount and cryptographic data
- Payment Gateway sends PED authorization response from issuing bank
- Card accepts authorization response and informs PED
- PED sends second set of cryptographic data to Payment Gateway
- PED prompts cardholder to remove card
- Payment Gateway sends response to POS
- POS prints transaction receipt
- Use case terminates

Preconditions
- None

Postconditions
- None

7   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 August 2010, and they may change over time. Integrators should consult with industry sources to get up-to-date information.

7.1   Lodging Interchange Requirements

7.1.1   Visa

- Magnetic Stripe or EMV-read
- Signature obtained for Magnetic Stripe-read
- Electronically authorized
- The transaction must be settled within 48 hours (2 days) 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

7.1.2   MasterCard

- Magnetic Stripe or EMV-read
- Signature obtained for Magnetic Stripe-read
- Electronically authorized
- The transaction must be settled within 48 hours (2 days) after the check-out

7.2   Restaurant/Retail Interchange Requirements

7.2.1   Visa

- Magnetic Stripe or EMV-read
- Signature obtained for Magnetic Stripe-read
- Electronically authorized
- The transaction must be settled within 48 hours (2 days) of the authorization

7.2.2   MasterCard

- Magnetic Stripe or EMV-read
- Signature obtained for Magnetic Stripe-read
- Electronically authorized
- The transaction must be settled within 48 hours (2 days) of the authorization

7.3   eCommerce Interchange Requirements

7.3.1   Visa

- Electronically authorized
- The transaction must be settled within 72 hours (3 days) of the authorization
- Authorization amount must equal transaction amount

7.3.2   MasterCard

- Electronically authorized
- The transaction must be settled within 72 hours (3 days) of the authorization

8   Messages

8.1.1   Request Data Element Table

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
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	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	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	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	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	Req
@Target	Used to indicate whether the request is for the Test or Production system.	Opt	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	Req
@TerminalID	This is the electronic address of the device from which information is entered.	Req	Req	Req	Req	Req	Req	Req	Req

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
@AgentSine	The user id of the clerk/server/cashier processing the transaction.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRQ / 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	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	Opt
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	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	Req
HTNG_PaymentCardProcessingRQ / AuthorizationDetail	Information with which the traveler's identification is verified and/or charges are authorized.	Req	Req	Req	Req	Req	Req	Req	Req
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization	Specifies credit card information about the customer seeking authorization.	Req	Req	Req	Req	Req	Req	Req	Req
@TransactionType	Authorization, Incremental Authorization, Auth Reversal, Settlement, Sale, Void Settlement, Return, Balance Inquiry	Req	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	Opt
@AuthorizationCode	This is the approval code received on the original authorization request. Used in cases where the requested transaction is to reverse the authorization or when a voice authorization is performed.	Opt	N/A	N/A	Req	N/A	N/A	N/A	N/A
@CardPresentInd	When true, the customer has actually presented the credit card.	Req	Req	Req	Req	Req	Req	Req	Opt
@Amount	A monetary amount.	Req	Req	Req	Req	Req	Req	Req	N/A

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
@AuthVerificationValue	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	N/A
@SourceType	Information describing the point of sale. Enumeration: NormalTransaction, MailOrPhoneOrder, UnattendedTerminal, MerchantIsSuspicious, eCommerceSecuredTransaction, eCommerceUnsecuredTransaction, InFlightAirPhone,CIS_NotLegible, CID_NotOnCard	Req	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	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard	Specifies the credit card information for which authorization is required.	Req	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	Cond Req
@CardNumberIsProxy	Indicates the value of @CardNumber is a Data Proxy value representing an actual credit card number.	Opt	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	Opt
@IsEMV	Indicated this is a Europay/Mastercard, Visa Chip & PIN enabled card.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@CardCode	The 2 character code of the credit card issuer.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
@ExpireDate	Indicates the ending date.	Cond Req	N/A	N/A	Cond Req	Cond Req	N/A	Cond Req	Cond Req
@EffectiveDate	For cards with a Start Date (typically Switch, Solo, and some Maestro cards.)	Cond Req	N/A	N/A	Cond Req	Cond Req	N/A	Cond Req	N/A

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
@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	N/A
@IssueNumber	For cards with an Issue Number (typically Switch, Solo, and some Maestro cards).	Cond Req	N/A	N/A	Cond Req	Cond Req	N/A	Cond Req	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard / CardHolderName	Name of the card holder.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / 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	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard / Address / AddressLine	May contain the street number and optionally the street name.	Cond Req	Opt	Opt	Opt	Cond Req	Opt	Opt	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / 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	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard / Address / PostalCode	Post Office Code number.	Cond Req	Opt	Opt	Opt	Cond Req	Opt	Opt	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail /	State or Province name (e.g., Texas).	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
CreditCardAuthorization / CreditCard / Address / StateProv									
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard / Address / CountryName	Country name (e.g., Ireland).	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard / MagneticStripe	Card Magnetic Stripe Data as defined by ISO 7813 for banking cards.	Cond Req	Opt	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	Opt
@Track2	The binary magnetic stripe data for track 2.	Cond Req	N/A	N/A	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / CreditCard / EMVData	Cryptograms and cardholder verification information required for Chip & Pin (EMV) transactions.	N/A	N/A	Cond Req	Cond Req	N/A	Cond Req	Opt	N/A
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / ID	Identification of an authorization party (e.g., merchant, acquirer).	Req	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). The OpenTravel value for Merchant ID is 32.	Opt	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	Req
HTNG_PaymentCardProcessingRQ / AuthorizationDetail / CreditCardAuthorization / ID / CompanyName	Identifies the company that is associated with the UniqueID.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
@CompanyShortName	Used to provide the company common name.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
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.). In the response message, this is typically used by the receiving system to determine how much gratuity was added to the check.	Opt	Opt	Opt	Req	Req	Req	Opt	N/A
@RevenueCategoryCode	Describes the type of revenue generated. Refer to OpenTravel Code List Revenue Category Code (RCC).	Opt	Opt	Opt	Req	Req	Req	Opt	N/A
HTNG_PaymentCardProcessingRQ / StayInfo / RevenueCategories / RevenueCategory / SummaryAmount		Opt	Opt	Opt	Req	Req	Req	Opt	N/A
@Amount	A monetary amount.	Opt	Opt	Opt	Req	Req	Req	Opt	N/A
HTNG_PaymentCardProcessingRQ / StayInfo / ReservationID	The confirmation number of the reservation associated with the stay.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
@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	N/A
@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	N/A
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
HTNG_PaymentCardProcessingRQ / StayInfo / HotelReservation	Contains the reservation details associated with the stay information.	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
@ResStatus	Indicates the status of the reservation.	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
@RoomStayReservation	Boolean True if this reservation is reserving rooms. False if it is only reserving services.	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
HTNG_PaymentCardProcessingRQ / StayInfo / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate	Provides information about the product the guest was sold.	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
@RatePlanCode	A string value may be used to request a particular code or an ID if the guest qualifies for a specific rate, such as AARP, AAA, a corporate rate, etc., or to specify a negotiated code as a result of a negotiated rate.	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
@RoomTypeCode	Specific system room type code, ex: A1K, A1Q etc.	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
HTNG_PaymentCardProcessingRQ / StayInfo / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate / Total	The total amount charged for the service including additional amounts and fees.	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
@AmountBeforeTax	The total amount not including any associated tax (e.g., sales tax, VAT, GST or any associated tax).	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
@AmountAfterTax	The total amount including all associated taxes (e.g., sales tax, VAT, GST or any associated tax).	Opt	N/A	N/A	Opt	Opt	N/A	N/A	N/A
HTNG_PaymentCardProcessingRQ / StayInfo / HotelReservation / RoomStays / RoomStay / TimeSpan	The Time Span which covers the Room Stay.	Req	Req	Req	Req	Req	Req	Req	N/A
@End	The ending value of the time span (e.g., Departure Date).	Req	Req	Req	Req	Req	Req	Req	N/A
@Start	The starting value of the time span (e.g., Arrival Date).	Req	Req	Req	Req	Req	Req	Req	N/A

8.1.2   *Response Data Element Table*

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
HTNG_PaymentCardProcessingRS	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	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	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.	Req	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	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	Req
@Target	Used to indicate whether the request is for the Test or Production system.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRS / 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	Opt
HTNG_PaymentCardProcessingRS / Warnings / Warning	Used in conjunction with the Success element to define a business error.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
@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	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	Req
@ShortText	An abbreviated version of the error in textual format.	Opt	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	Opt
HTNG_PaymentCardProcessingRS / Authorization / AuthorizationDetail	The original authorization request information.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRS / Authorization / AuthorizationDetail / CreditCardAuthorization	Specifies credit card information about the customer seeking authorization.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
@TransactionType	Authorization, Incremental Authorization, Auth Reversal, Settlement, Sale, Void Settlement, Return, Balance Inquiry	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
@SaleCode	Advance Deposit, Delayed Charge, Express Service, Assured Reservation, Normal Charge, No Show Charge.	Opt	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	N/A
@Amount	A monetary amount.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
HTNG_PaymentCardProcessingRS / Authorization / AuthorizationDetail / CreditCardAuthorization / CreditCard	Specifies the credit card information for which authorization is required.	Opt	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	Opt
@CardCode	The 2 character code of the credit card issuer.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
@ExpireDate	Indicates the ending date.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRS / Authorization / AuthorizationDetail / CreditCardAuthorization / CreditCard / EMVData	Cryptograms and cardholder verification information required for Chip & Pin (EMV) transactions.	Cond Req	N/A	Opt	Opt	Cond Req	Opt	Opt	N/A
HTNG_PaymentCardProcessingRS / Authorization / AuthorizationDetail / CreditCardAuthorization / ID	Identification of an authorization party (e.g., merchant, acquirer).	Opt	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	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	Opt
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
HTNG_PaymentCardProcessingRS / Authorization / AuthorizationDetail / BookingReferenceID	The booking or confirmation number for which charges are being authorized.	Opt	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	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	Opt
@ID_Context	Used to identify the source of the identifier (e.g., IATA, ABTA).	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Opt
HTNG_PaymentCardProcessingRS / Authorization / AuthorizationResult	Result information from the authorization process.	Req	Req	Req	Req	Req	Req	Req	Req
@ApprovalDateTime	The date and time that the approval was issued.	Opt	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	N/A
@AddressResultCode	The result of the address validation.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
@AuthorizationCode	The unique code returned by the authorizing party. This is returned for successful authorizations.	Opt	Opt	Opt	Opt	Opt	Opt	Opt	N/A
@Result	Information returned by the credit card vendor describing the results of the processing of the request.	Req	Req	Req	Req	Req	Req	Req	Req
@Description	Additional response information.	Req	Req	Req	Req	Req	Req	Req	Req
@ApprovedAmount	The amount that was able to be applied against the card. This amount may be different then the requested amount.	Req	Req	Req	Req	Req	Req	Req	N/A
@BalanceRemainin	For stored-value cards	Opt	Opt	Opt	Opt	Opt	Opt	Opt	Req

Element | @Attribute	Description/Contents	Authorization	Incremental Auth	Auth Reversal	Settlement	Sale (No Prior Auth)	Void Settlement	Return	Balance Inquiry
g	and gift cards, this is the value of the card.								

8.2   Sample Messages

8.2.1   Authorization

Request

```xml
<?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
<?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>
```

8.2.2   Incremental Authorization

Request

```xml
<?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
<?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>
```

8.2.3   Authorization Reversal

Request

```xml
<?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>
```

8.2.4   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>
```

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>
```

8.2.5   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>
```

8.2.6   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>
```

8.2.7 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>
```

8.2.8 Balance Inquiry

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="Balance Inquiry"
SourceType="NormalTransaction">
                        <CreditCard CardNumber="7500316746674444"/>
                        <ID Type="10" ID_Context="PMTGTYABC" ID="10462786704"/>
                </CreditCardAuthorization>
        </AuthorizationDetail>
</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="Balance Inquiry">
                                <CreditCard MaskedCardNumber="XXXXXXXXXXXX1111" ExpireDate="0212"/>
                                <ID Type="10" ID_Context="PMTGTWABC" ID="10462786704"/>
                        </CreditCardAuthorization>
                </AuthorizationDetail>
                <AuthorizationResult ApprovalDateTime="2010-02-12T01:21:55" Result="Approved"
Description="Balance Inquiry successful" BalanceRemaining="100.00"/>
        </Authorization>
</HTNG_PaymentCardProcessingRS>
```

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

9.1   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 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/

9.2   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

9.3   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 are as follows:

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.

9.4   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).

9.5   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