

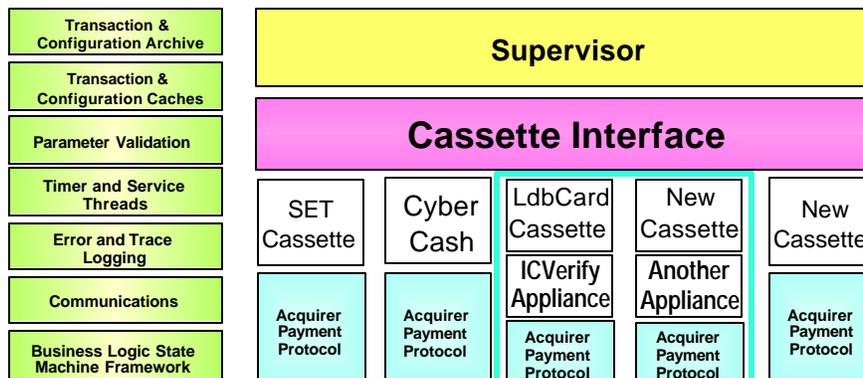
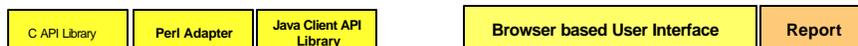
# IBM Payment Manager Cassette Development Workshop

Payment Objects, States, and  
Commands



Angelo Scaccabarozzi  
Merchant Enablement  
Team

## Payment Manager Internals



## Framework Payment Objects

- ORDER
  - ▶ Represents shopper's intention to pay for a purchase with a single payment instrument.
  - ▶ Contains information for identifying shopper
  - ▶ Typically, one-to-one relationship with merchant server order
  - ▶ Data and state stored in ETORDER Table

## Framework Payment Objects

- PAYMENT
  - ▶ Money transferred from shopper to merchant
  - ▶ More than one payment may exist for an order
  - ▶ Typically, must be authorized by payment network
  - ▶ Collected in a Batch when goods are shipped
  - ▶ Data and state stored in ETPAYMENT Table

## Framework Payment Objects

- CREDIT
  - ▶ Money returned to the shopper (e.g. refund for returned goods)
  - ▶ Zero or more credits per order
  - ▶ Collected in Batch when refund is made
  - ▶ Data and state stored in ETCREDIT Table.

## Framework Payment Objects

- BATCH
  - ▶ Processes a group of payments and credits
  - ▶ Batch closure triggers the transfer of money
  - ▶ Data and state stored in ETBATCH Table
  - ▶ Many types can be chosen by cassette
    - Implicit open: Cassette opens as needed
    - Explicit open: Application calls BatchOpen
    - Other variations

## **Framework Object State**

- BATCH, ORDER, PAYMENT and CREDIT objects each have defined states.
- State changes as a result of a command.
- State determines which commands can legally operate on an object.
- Cassettes manage the state of Framework objects - only cassettes can understand how a framework object state must be changed.

## **Framework Object state (*continued*)**

- Cassette checks object state to assure operation is legal.
- Cassette updates a state as necessary.

# Payment API Commands

- RECEIVEPAYMENT
  - ▶ Creates an ORDER object.
  - ▶ Order information is provided by merchant
  - ▶ Payment information is taken from shopper's wallet (a Browser plug-in)
  - ▶ Protocol between wallet and cassette is cassette specific
  - ▶ Autoapprove and autodeposit flags to automatically process approve and deposit.
  - ▶ After successful completion the order state will be set to ORDERED or REFUNDABLE

# Payment API Commands

- ACCEPTPAYMENT
  - ▶ Creates an ORDER object.
  - ▶ Order information and payment information are provided by merchant (usually taken from HTML form)
  - ▶ Autoapprove and autodeposit flags to automatically process approve and deposit.
  - ▶ Specific cassette payment data provided via Protocol Data . For instance:

key	value
\$BRAND	VISA
\$PAN	110010873
\$EXPIRY	199912

- ▶ After successful completion the order state will be set to ORDERED or REFUNDABLE

# Payment API Commands

- APPROVE
  - ▶ creates a PAYMENT object
  - ▶ approve request for the total or partial amount purchase
  - ▶ the cassette will forward the request to payment appliance / processor
  - ▶ no money is transferred from buyer to merchant.
  - ▶ in case of split payment, multiple approvals are allowed but the total amount must not exceed the purchase amount.
  - ▶ associated order must be in ORDERED or REFUNDABLE state
  - ▶ After successful completion the state of payment object will be set to APPROVED.
  - ▶ If not approved, state becomes DECLINED

# Payment API Commands

- APPROVEREVERSAL
  - ▶ Replace the amount of an approval with the new amount of the approve reversal
    - Correct errors
    - Refresh expired approval
    - Split payment after single approval
    - Cancel (void)
  - ▶ The cassette will forward the request to payment appliance / processor
  - ▶ Associated payment must be in APPROVED state
  - ▶ In case of total reversal payment is set to VOID, otherwise APPROVED

## Payment API Commands

- DEPOSIT
  - ▶ Order or suborder has shipped and funds must be captured
  - ▶ The cassette will forward the request to payment appliance / processor
  - ▶ Money transfer from buyer to merchant is prepared (money will be actually transferred when batch is closed).
  - ▶ Associated payment must be in APPROVED state and will be set to DEPOSITED after successful completion.

## Payment API Commands

- DEPOSITREVERSAL
  - ▶ Set deposit amount to zero (in other words, remove deposit from batch)
    - To correct errors
    - Cancel (void)
  - ▶ The cassette will forward the request to payment appliance / processor
  - ▶ Associated payment must be in DEPOSIT state
  - ▶ Payment is set to APPROVED after successful completion

## Payment API Commands

- REFUND
  - ▶ A CREDIT object is created.
  - ▶ Associated order must be in REFUNDABLE state.
  - ▶ A Refund request is sent to the cassette and then processed by payment appliance / processor
  - ▶ Money transfer from merchant to buyer is prepared (money will be actually transferred when batch is closed).
  - ▶ Credit is set to REFUNDED after successful completion

## Payment API Commands

- REFUNDREVERSAL
  - ▶ Set refund amount to zero (in other words, remove refund from batch)
    - To correct errors
    - Cancel (void)
  - ▶ The cassette forwards the request to payment appliance / processor
  - ▶ associated credit must be in REFUNDED state
  - ▶ the credit is set to VOID after successful completion

## Payment API Commands

- BATCH Open
  - ▶ Used by merchant to explicitly open a new batch (not a common practice)
  - ▶ Not supported by User Interface
  - ▶ Use implicit batch open instead
  - ▶ A Batch object is created
  - ▶ After successful completion the batch is set to OPEN and status is set to Not Yet Balanced

## Payment API Commands

- BATCHCLOSE
  - ▶ Used by merchant to explicitly force batch settlement
  - ▶ Cassette forwards request to payment appliance / processor.
  - ▶ After successful completion:
    - the batch is set to CLOSED.
    - the payments and credits contained in the batch are set to CLOSED.
  - ▶ Batch has also a status reflecting balance state (not yet balanced, balance closed and balance failed).

# Commands Mapped to Actions

- Commands are defined with universal terms that could loosely apply to every payment type.
- Mapping between commands and your payment processing is instrumental to develop a new cassette

Merchant SW action	Credit Card Action	Payment Mgr command
Book	Authorize	Approve
[Book] Void	Void	Approve Reversal
Ship	Capture	Deposit
[Ship] Void	Void	Deposit Reversal
Credit	Credit	Refund
[Credit] Void	Void	Refund Reversal
Collect	Settle	Close Batch