Attachment C

to

RFP No. 4666

Mississippi Department of Information
Technology Services

Avaya Products, Services, and CCaaS

Interfacing with Tyler Technologies

ITS Project No. 48690

Tyler Technologies d/b/a Tyler Mississippi (previously Mississippi Interactive and NIC Mississippi) will serve as the single point of entry for all e-commerce and card present (POS) transactions. Awarded vendor must use Mississippi's official payment processor, Tyler MS, for any of the following services where payment is required.

- TPE Payment Engine gateway
- Common Checkout Page (CCP) module for online web and mobile payment pages
- Over the Counter (OTC) module for Point-of-Sale transactions
- Secure Wallet module to support scheduled and recurring payments
- Devices Client module for managing the secure communications between credit card terminals and point-of-sale solutions

The following payment methods accepted through Tyler MS include: Visa, MasterCard, American Express, Discover, electronic check and subscription (monthly billed). Tyler MS requires that prior to sharing any integration guides or materials the vendor execute a non-disclosure agreement with Tyler MS.

DFA Administrative Rule

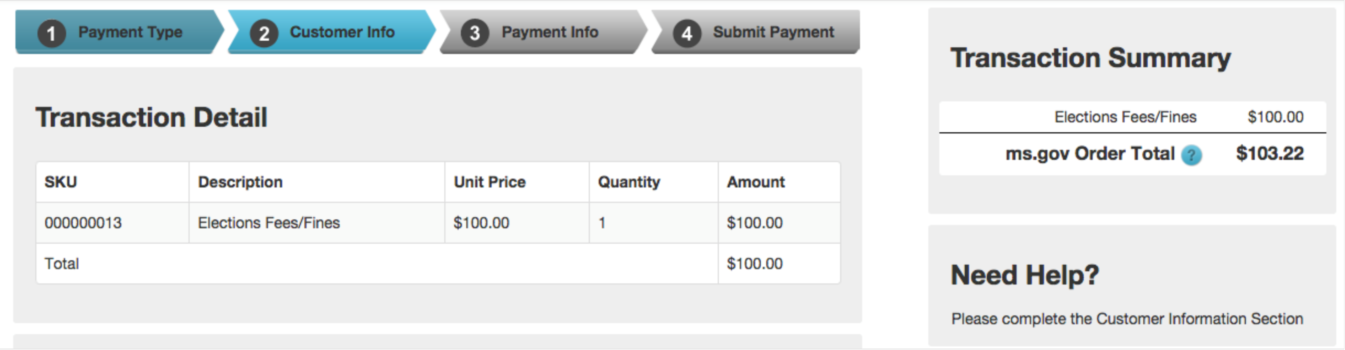
The Department of Finance and Administration (DFA) established an administrative rule to be followed when agencies, in accordance with §27-104-33, Mississippi Code of 1972, Annotated, elect to accept payment by credit cards, charge cards, debit cards, electronic check (echeck) and other form of electronic payments for various services and fees collectible for agency purposes. See Attachment B for Final Rule.

Payment Card Industry (PCI) Compliance

Tyler MS will be responsible for reporting Payment Card Industry (PCI) compliance on behalf of the State. Each Agency will have some requirements that they are responsible for depending upon the payment service selected. These requirements will be communicated with each Agency prior to the start of service, though any future change in the PCI standards may require additional support from the State entity and awarded vendor. Tyler MS payment processing services, including but not limited to the Transaction Processing Engine (TPE), are certified compliant with the PCI Data Security Standard (DSS) as a Level 1 Service Provider. Tyler is also listed as a Validated Payment Service Provider by VISA and MasterCard. TPE is hosted at a Central Data Center in Ashburn Virginia and complemented with a backup facility in Allen, Texas.

See Technical Requirements for notes to the PCI compliance responsibility of the awarded vendor.

The fee break out can include a "subtotal" for services and a "Total ms.gov Price" or "ms.gov Order Total" which includes the eGov processing fee. See image below for example.



Transaction Detail

SKU	Description	Unit Price	Quantity	Amount
000000013	Elections Fees/Fines	$100.00	1	$100.00
Total				$100.00

Transaction Summary

Elections Fees/Fines	$100.00
ms.gov Order Total ?	$103.22

Need Help?

Please complete the Customer Information Section

Merchant of Record

In order to act as the single point of contact between the State, Tyler MS, the payment processor, the merchant acquiring bank, and end users of ms.gov services, Tyler MS will be the “Merchant of Record” for this RFP. As the single point of contact for the State, Tyler MS will work directly with the processor and the acquiring bank to request and set up merchant accounts and will be responsible for all areas of merchant services, including merchant fees.

eGov Transaction Fees

There will be standard payment processing fees associated with each payment transaction. Customer approval (electronic or otherwise) of Tyler MS payment processing fees will be obtained prior to initiating payment.

MAGIC

Tyler MS payment solution processes is integrated with MAGIC, Mississippi’s statewide accounting and procurement system of record. At least three (3) weeks prior to service launch, the Mississippi agency will be required to work with DFA to set up corresponding charges table entries. After appropriate edits are made to the charges table, the Mississippi agency seeking services and awarded vendor will be required to work with Tyler MS to ensure adequate testing, confirming the application transactions are posting to MAGIC. A live transaction test must be completed no later than three (3) business days before service launch.

Refunds, Chargebacks, Returns

As the merchant of record and official payment processor, Tyler MS will handle all refunds, chargeback representments and returned echecks. However, Tyler MS is not responsible for covering any monies that must be netted from the agency’s account through refund, successful chargeback or returned echeck. Below are the processes for each.

<u>Refunds</u>

The refund process is initiated by either customer or agency request.

- Upon customer request Tyler MS will contact the agency financial contact (established at project initiation) for approval prior to refund.
- Agency contacts have access to and are encouraged to use the Tyler MS refund tool for their refund requests. This ensures adequate logs of all requested refunds
- Refund API is available for programmatic refunds and approval of use is required by the individual agency.
- After agency request or approval, Tyler MS refunds the charge in TPE and notifies the requestor upon completion.

Through MAGIC refunds are netted from the next day's deposits, or the next deposit containing funds for the refund to net.

Chargebacks
A chargeback is a monetary dispute that is initiated by the Issuing Bank (issuer disputes the posting of the transaction) or the cardholder (a cardholder disputes a transaction).

- Customer or card issuing bank sees what appears to be a suspicious charge on their statement.
- The customer contacts the card company to dispute the charge and initiate the chargeback process. Note: depending on the company policies of the company that issued the card the company may initiate the chargeback without customer notification.
- Tyler MS receives a chargeback email from our processor notifying us of the transaction details of the chargeback. Once this notification is received the processor pulls the funds back from the Portal account until supporting documentation is obtained. (Tyler MS' processor has 45 days from the time the customer disputes the charge to contact Tyler MS for additional information.)
- Based on the information provided in the chargeback notification, Tyler MS researches the charge internally first.
- If the disputed charge is a true duplicate charge (same customer information, amount, etc.), Tyler MS allows the chargeback to process and it is automatically marked in TPE. In the event that Tyler MS needs agency verification, Tyler MS contacts the appropriate agency contact(s) (financial contact gathered at project initiation) by email to explain the chargeback, provide charge details, and verify with the contact that it is a valid charge. If needed, Tyler MS requests the agency provide any additional information they may have to support the claim.
- If the charge is valid, Tyler MS will provide the sales drafts (chargeback receipt, TPE receipts, agency support, etc.) back to the processor to support the charge validity.
- If the agency advises that the chargeback should result in the return of funds to the customer, Tyler MS will mark the chargeback as accepted.
- After the charge is verified through receipt of sales drafts, the chargeback will be reversed and the funds will be deposited back to the agency.
- In the event the issuing financial institution reviews the documentation and agrees with the customer, a second chargeback will be issued to recoup the funds and the case will be closed. The agency is notified of the chargeback and can handle collection of funds as they see fit.

Note: The chargeback process could take up to 60 days to resolve.

Returns
Electronic checks (echeck)/ACH payments (where a user enters an account and routing number) may be returned unpaid for any reason, including non-sufficient funds (NSF), stop payment, online data entry error, or closed account. A full list of return codes is listed below:

- R01 - Insufficient Funds - Available balance is not sufficient to cover the dollar value of the debit entry.
- R02 - Account Closed - Previously active account has been closed by customer or RDFI.
- R03 - No Account/Unable to Locate Account - Account number structure is valid and passes editing process but does not correspond to individual or is not an open account.
- R04 - Invalid Account Number - Account number structure not valid; entry may fail check digit validation or may contain an incorrect number of digits.
- R05 - Improper Debit to Consumer Account - A CCD, CTX, or CBR debit entry was transmitted to a Consumer Account of the Receiver and was not authorized by the Receiver.
- R06 - Returned per ODFI's Request - ODFI has requested RDFI to return the ACH entry (optional

to RDFI – ODFI indemnifies RDFI).

- R07 - Authorization Revoked by Customer - Consumer, who previously authorized ACH payment, has revoked authorization from Originator (must be returned no later than 60 days from settlement date and customer must sign affidavit).
- R08 - Payment Stopped - Receiver of a recurring debit transaction has stopped payment to a specific ACH debit. RDFI should verify the Receiver's intent when a request for stop payment is made to ensure this is not intended to be a revocation of authorization.
- R09 - Uncollected Funds - Sufficient book or ledger balance exists to satisfy dollar value of the transaction, but the dollar value of transaction is in process of collection (i.e., uncollected checks) or cash reserve balance below dollar value of the debit entry.
- R10 - Customer Advises Not Authorized - Consumer has advised RDFI that Originator of transaction is not authorized to debit account (must be returned no later than 60 days from settlement date of original entry and customer must sign affidavit).
- R11 - Check Truncation Entry Returned - used when returning a check safekeeping entry; RDFI should use appropriate field in addenda record to specify reason for return (i.e., "exceeds dollar limit," "stale date," etc.).
- R12 - Branch Sold to Another DFI - Financial institution receives entry destined for an account at a branch that has been sold to another financial institution.

Typical Return Process

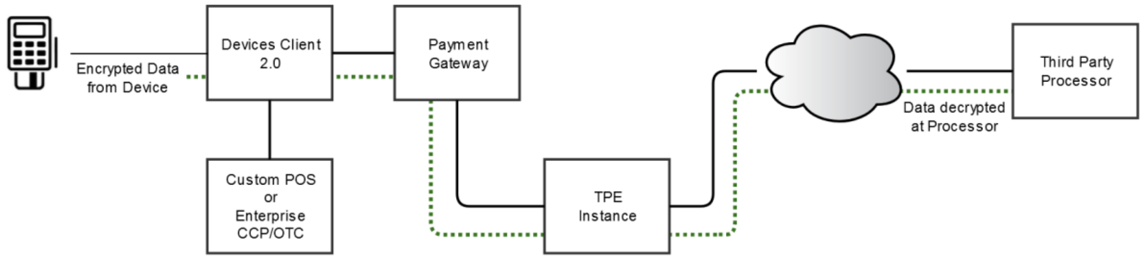
- User enters echeck information in the ms.gov common checkout page
- TPE captures the information and sends to payment service provider
- The service provider submits a request to the payer’s bank to retrieve the funds
- Payer’s bank reports back one of the aforementioned return codes to the services provider
- Service provider notifies Tyler MS and the return is marked in TPE
- Funds are electronically pulled from the agency through the daily MAGIC payment interface file. Tyler MS contacts the individual(s) responsible for agency funds (contact obtained during project initiation) by email to let them know of the return and reason. From there, the agency can handle collection of funds as they see fit.

Hardware Acquisition

Due to the payment key injections required for hardware to be compatible with Tyler MS’ PCI compliant payment processor, any hardware must be acquired through Tyler MS’ existing eGov contract. This includes, but is not limited to, kiosks, pin pad/card swipe, mobile devices, etc.

Devices Client

Devices Client allows the calling application to send the order information to Devices Client which will then interact with the payment device, add payment information to the order, and pass the order to the back end for processing. This will allow the calling application to be removed from PCI scope as none of the payment information will be passed through it. It will also allow for easier EMV integrations for new services since the EMV flow has already been certified and will only need to do regression tests for new certification.


Encrypted Data from Device
Devices Client 2.0
Custom POS or Enterprise CCP/OTC
Payment Gateway
TPE Instance
Data decrypted at Processor
Third Party Processor

Application Testing
For all new services, DFA requires a test transaction to be run for flow of funds and processor verification. After Tyler MS receives confirmation the awarded vendor is satisfied with the integration, one test must be run through production TPE and confirmed by Tyler MS.

It takes three (3) business days (excluding bank holidays) for the transaction to be confirmed by DFA. Awarded vendor should take this time frame into consideration when anticipating launch date.

Reporting
TPE provides reporting and auditing tools useful for streamlining and accommodating various back-office procedures. TPE's financial reporting is comprehensive, flexible, and robust. Within TPE all payment processing data is made available via a wide variety of reporting features. Reports are real-time, up-to-the-minute transaction reporting ranging from summary reports to detail reports showing line-item level data. A comprehensive users guide and applicable training will be provided to agency contacts during integration.

Payment Support
Tyler MS will provide support for all user payment inquiries. Tyler MS is located at 2727 Old Canton Road, Suite 100, Jackson, Mississippi 39216 and customer payment support is available during normal business hours (Monday – Friday 8:00 a.m.-5:00 p.m. CST). Tyler MS' toll free support number (1-877-290-9487) is listed on the ms.gov Common Checkout page and is accessible to all users. For payment emergencies a technical support cellular number will be provided to the State contact.

Tyler MS will work directly with the awarded vendor and/or the agencies to identify, report, track, monitor, escalate, and resolve any technical issues with TPE or CCP. It is Tyler MS' policy to notify all awarded vendors and agencies of planned maintenance windows or system updates to avoid any payment issues.

State entities and/or awarded vendors will not be charged for Tyler MS' efforts during payment implementation or any training/support.

Technical Requirements
Mississippi's payment solution is designed to provide two methods of integration: CommonCheckout (where the customer clicks on a "Pay Now" button and is transferred to a set of common checkout pages branded for ms.gov), and DirectConnect (where the application has self-contained checkout pages and will call TPE for verification and capture once all payment information has been entered). In both of these instances, the awarded vendor will utilize standard web service protocols.

The CommonCheckout integration is required by ITS and DFA. Should special circumstances arise where the CommonCheckout is not applicable and/or the DirectConnect option is required, approval from both State agencies is mandatory.

High-level descriptions of the integration requirements are included in this section. For detailed documentation please contact David Campbell, Tyler MS' Director of Technology, at david.j.campbell@tylertech.com.

CommonCheckout (CCP)

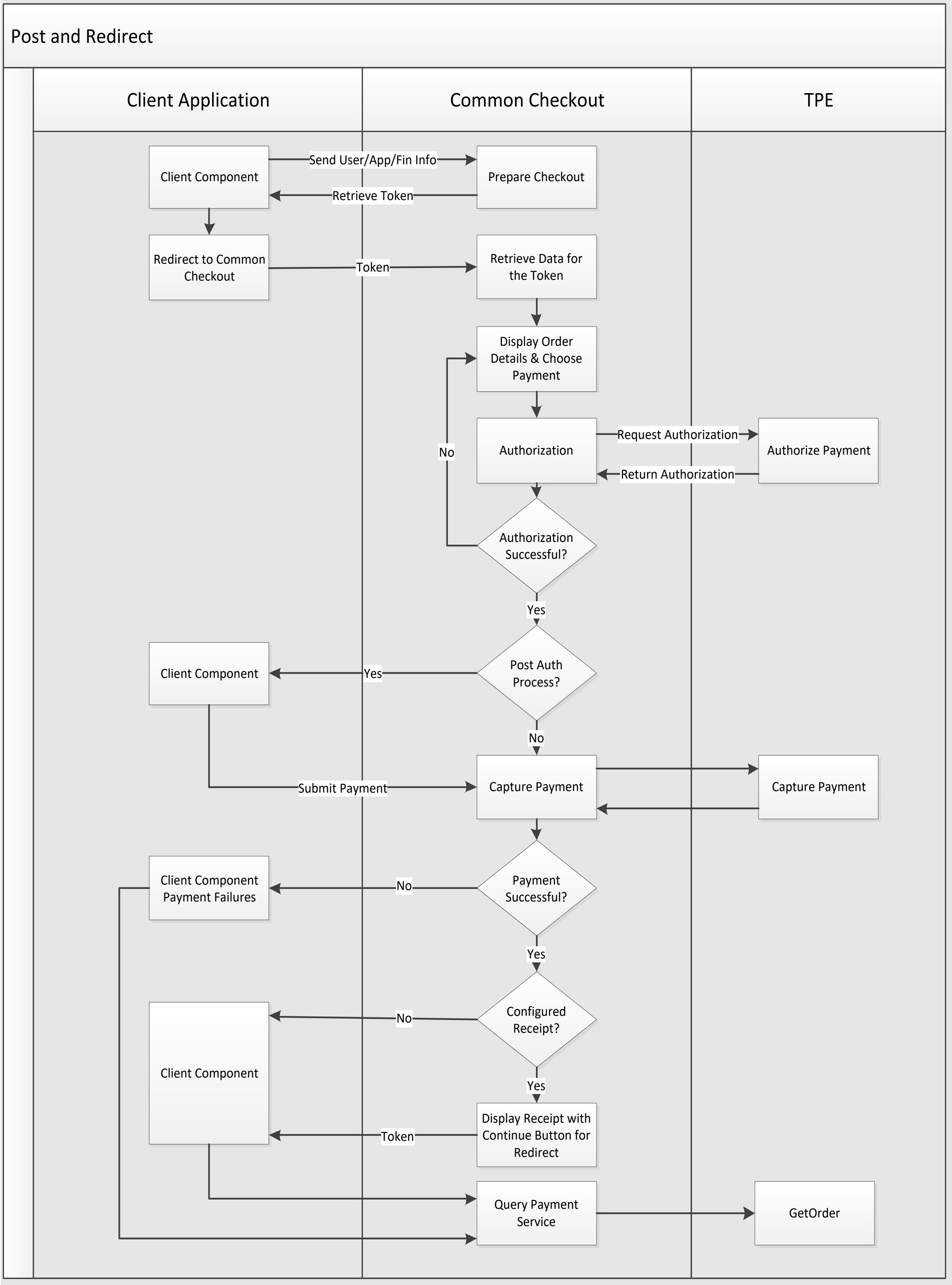
When utilizing CommonCheckout, the calling application is not responsible for collecting the credit card or banking information. Instead, the application sends the transaction data to the CommonCheckout interface which collects and processes all payment information. The CommonCheckout interface will then return to the calling application all transaction status details and information related to the transaction.

CCP Option 1: Server-side Web Service Calls and Browser-side Redirect

To initiate a transaction, the partner application is required to invoke the PreparePayment method on the Common Checkout web service (ReST or SOAP) that is passing along the financial/customer/application information.

- The Web Service operation returns a token back in the SOAP or ReST response. This token allows CCP to recall and process the transaction using the information provided during the PreparePayment call.
- This token is required as a query string parameter when redirecting the user to CCP for collection and processing of payment information.
- When the customer chooses to continue with the payment by clicking a form button on the partner screen, the browser redirects to the Common Checkout web application.
- The Common Checkout web application retrieves the customer/financial/application data associated with the token and displays it on the payment page.
- Upon submission of the payment, Common Checkout redirects to the partner application or displays a receipt page, based on the configuration. In the latter case, the redirect to the partner application happens when a customer clicks a button on the receipt screen.
- The partner application is required to do a call back to the Query payment web service by sending the token. The service will return the transaction information back in the SOAP or Rest response. Optionally, the partner application may receive information about the payment via a 'postback' message from CCP. These methods ensure the authenticity of the payment.

  The following figure outlines a typical process flow for a CommonCheckout transaction.


Post and Redirect
Client Application
Common Checkout
TPE
Client Component
Send User/App/Fin Info
Prepare Checkout
Retrieve Token
Redirect to Common Checkout
Token
Retrieve Data for the Token
Display Order Details & Choose Payment
Authorization
Request Authorization
Authorize Payment
Return Authorization
No
Authorization Successful?
Yes
Client Component
Yes
Post Auth Process?
No
Submit Payment
Capture Payment
Capture Payment
Client Component Payment Failures
No
Payment Successful?
Yes
Client Component
No
Configured Receipt?
Yes
Token
Display Receipt with Continue Button for Redirect
Query Payment Service
GetOrder

CCP Option 2: Server-side Web Service Calls and InContext IFrame Display
Partners who do not wish to redirect users away from and wish to display payment options on the page may utilize the CCP InContext Iframe Display view which allows Common Checkout to be embedded within the Partners page via an Iframe.

To initiate a transaction, the partner application is required to invoke the PreparePayment method on the Common Checkout web service (ReST or SOAP) that is passing along the financial/customer/application information.

- The Web Service operation returns a token back in the SOAP or ReST response. This token allows CCP to recall and process the transaction using the information provided during the PreparePayment call.
- This token is required as a query string parameter when redirecting the user to CCP for collection and processing of payment information.
- When the customer chooses to continue with the payment by clicking a form button on the partner screen, the partner's application call renders the Common Checkout web application embedded in an Iframe on the current page.
- The Common Checkout web application retrieves the customer/financial/application data associated with the token and displays it on the payment page within the Iframe.
- The Common Checkout application within the Iframe communicates to the partner application on the parent window via the application subscribing to an event listener and receiving a JSON response.
- Upon submission of the payment, Common Checkout provides a JSON event response informing the partner application of the status of the payment.
- The partner application can choose to close the Iframe and build and display its own receipt.
- The partner application is required to do a call back to the Query payment web service by sending the token. The service will return the transaction information back in the SOAP or Rest response. Optionally, the partner application may receive information about the payment via a 'postback' message from CCP. These methods ensure the authenticity of the payment.

Wallet API One-Time and Stored Payments

The Wallet API provides payment integration services for front-end web applications through a Rest interface. It enables front-end application users to save credit card and bank account information and complete payment transactions with the saved payment accounts. Payment account data is stored in a PCI/PII compliant account vault. Communication with the API is restricted to authorized application hosts. Partner application users do not interact directly with the API.

To initiate Guest checkout transactions:

- The partner application calls the Wallet API to authorize caching a payment account.
- The Wallet API returns a URL for an IFrame to be hosted on a partner application web page.
- Customers enter credit card or e-checking data in the iFrame, keeping the front-end application out of PCI scope.
- Cached payment accounts are referenced by the Wallet API for payment transactions routed to TPE.
- TPE returns the payment result to the Wallet API, which passes the result to the Partner Application as a JSON response.
- The partner application displays the payment result to the end user.

- Cached payment accounts can be saved if desired.

To initiate saved payment account transactions:

- The partner application calls the Wallet API to authorize saving a payment account.
- The Wallet API returns a URL for an IFrame to be hosted on a partner application web page.
- Customers enter credit card or e-checking data in the iFrame, keeping the front-end application out of PCI scope.
- Saved payment accounts are referenced by the Wallet API for payment transactions routed to TPE.
- TPE returns the payment result to Wallet API which passes the result to the Partner Application as a JSON response.
- The payment method is encrypted and saved in a PCI compliant vault.
- Saved payment accounts are referenced by Wallet API user IDs, shared with the partner application.

DirectConnect

The second scenario is to use the Application Programming Interfaces ("API's") that are available to developers. In this scenario, agency or third party developers write applications that include the checkout pages. Customers fill out all payment information within the application, and once captured, the application communicates with TPE using a standard ReST API. TPE processes the payment, based on payment type, and returns either a success or failure code back to the calling application. Based on the code, the calling application displays either a receipt back to the customer or the reason for the failure. TPE supports multiple API languages including:

- Java
- .NET
- Perl
- PHP

Note: If the DirectConnect method is approved by ITS and DFA the awarded vendor must provide Tyler MS and the State proof of their software's (and any applicable hardware used for hosting the software) PCI compliance through a SAQ-D service provider version or a PCI Attestation of Compliance (AoC) completed by a Qualified Security Assessor (QSA). Approved Scanning Vendor (ASV) quarterly application scans will also be required to be submitted to Tyler MS.

DirectConnect Integration Outline: Before a payment can be processed inside of TPE, there are preliminary steps needs to happen. Couple of setup needs to happen before an order can be made. This setup includes:

- Processor Setup
- Service Code Setup
- Merchant Setup

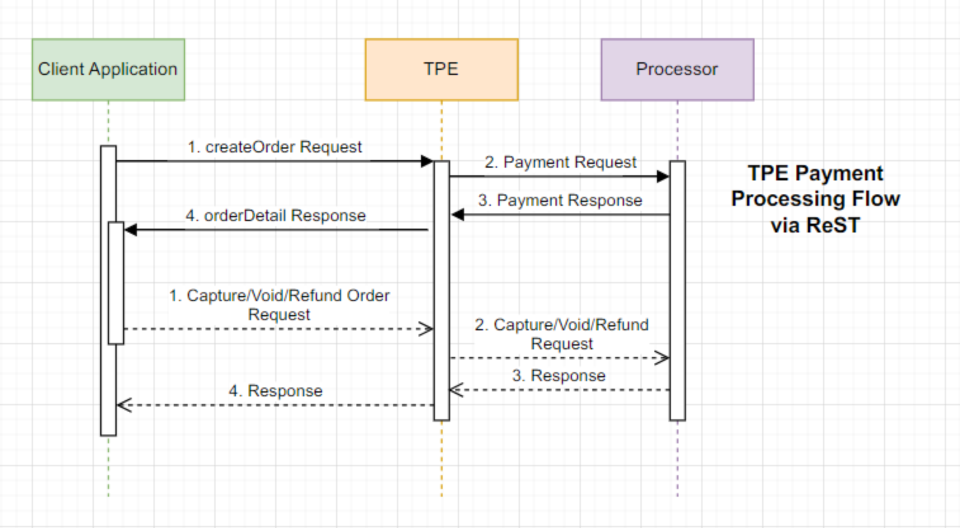
This setup can be done via Tyler representative. Once this setup is done, the service code becomes the key element for all order requests. An Order is the basic transaction container in TPE. It is a detailed request for certain goods or services and represents all the instructions and information needed from the customer for the merchant to collect money. An order contains information about the customer, items purchased, fees and taxes, payment information, billing address, shipping address, and so forth.

TPE uses the term order, along with the terms payment and credit to represent payment data for all electronic payments. An order is created by the client application while the customer is placing an order for goods or services. Transactions flow between the merchant and the financial institution during the life cycle of the order. These transactions can be broken into two broad categories: payments (monies transferred to the merchant from the customer) and credits (monies returned to the customer, such as when goods or services are returned and payment is refunded). As order processing continues, payments and credits are created and modified.

The basic steps for creating an Order and processing a payment are as follows:

1. **Submit a new Order Request to TPE:** The client application will create a request that includes a Merchant Id, a Merchant Key, and a Service Code. Along with this preliminary information, the client application will need to send these following fields which represents the payment:
   - Payment Account
     - Type of transaction: ecomm, retail, etc
     - Action: sale, authorization, refund, etc
     - Billing Address
     - CardData: Credit Card, ACH, Cash, etc
   - Items
     - This will have all the payment items which are part of the transaction. This represents the transaction line items and amounts.
2. When the Order call is being made by calling application, it will be submitted for authorization. TPE will do preliminary validations on the Order before submitting it to the Merchant Service Provider for authorization. If there is an error with the Order, TPE will return that information back to client application, or it will return back that the authorization was successful.
3. **Complete the Order:** Once the authorization is completed, the client application can use the Order ID from TPE to complete the order. This call to TPE informs the system that the order is complete and ready to be invoiced. In this call, the calling application informs TPE to either capture or void the order. Once this is done, the money transfer (i.e., Capture) is initiated. The invoice takes the information from the Order, and is then submitted to the Merchant Service Provider for Capture/Settlement. To have the invoice and order completion with this call itself, the service code needs to be setup accordingly.

The following figure outlines a typical process flow for a Direct Connect transaction.



<u>Charges Use in Tyler MS Common Checkout</u>

The ChargeItem data will become the basis for a line item that is sent to the CCP in the Prepare Checkout call. The table below maps the line item fields referenced in the CCP interface to their related ChargeItem value. In the CCP Prepare Checkout service call, line items are sent in as an array of lineItems.

CCP Line Item element	**Field Description**	**Field used from Charges Item**
LineItem.SKU	Item identifier used in backend SAAS funds distribution.	ChargeItem.itemType
LineItem.Description	Description of the item being purchased.	ChargeItem.description
LineItem.Unit_Price	Cost of 1 of this item.	ChargeItem.amount
LineItem.Quantity	Quantity of the item being purchased.	Computed by the application.