RFP 16-01

EXHIBIT M

Corporations and Charities System

Detailed Design Document

April, 2015

TABLE OF CONTENTS

Document Revision History

Version	Date	Description
	4/02/2015	Drafted by Sanjeev Batta presented at meeting

1. INTRODUCTION

1.1 Purpose

This document describes the detailed design for the Corporations and Charities System. The detailed design document describes the following

- A Technology Model / Framework for Rest Services.
- A draft set of Rest Services exposed by the system to fulfill the design requirements for web interface as well as staff console. The draft or proposed is used to describe these services because actual development of these services in an iterative and agile fashion will introduce changes along with an improved understanding of customer, system and performance needs that will further help refine and finalize these services.
- Details on integration specifications in some of the logical design areas where further definition / documentation can improve the understanding of data model and approach to integration with these systems.

1.2 Scope

The scope of this document is limited to corporations and charities system detailed design. The changes to other systems e.g. Combined Fund Drive, revenue etc. are not included in this design.

2. TECHNOLOGY MODEL

The technology component of the major section of the project is defined in the logical architecture. This model is to elaborate on specific areas of technology to further clarify the intent of the logical architecture. Certain detailed design items will be clarified and defined in an agile development lifecycle based on the basic framework highlighted here in.

The Rest Services for the project will be developed using .Net WebAPI. The design leverages the WebAPI controllers as a façade layer and uses a business domain object layer that provides a certain set of operations.

The REST Services layer has JSON / Parameter input and returns either the HTTP status codes with success based URI's or returns a JSON response. JSON response format used by the framework is based on JSend. JSend allows a consistent way of handling different UI errors as well other features to enable development of solid rest interfaces.

The WebAPI handlers for each of the different services are nested with the WebAPI routes and handle the defined service calls. The WebAPI handlers implement the overall business rules validation for each service, security validation and other aspects of Transaction and rules validation.

Each Business Domain Service is broken down into different domain layer classes that handle the domain specific logic as well as abstract the data access for the application. E.g. Order Domain Layer use SQL Data provider while the Filing / Corporations domain layer uses Mongo for persistence.

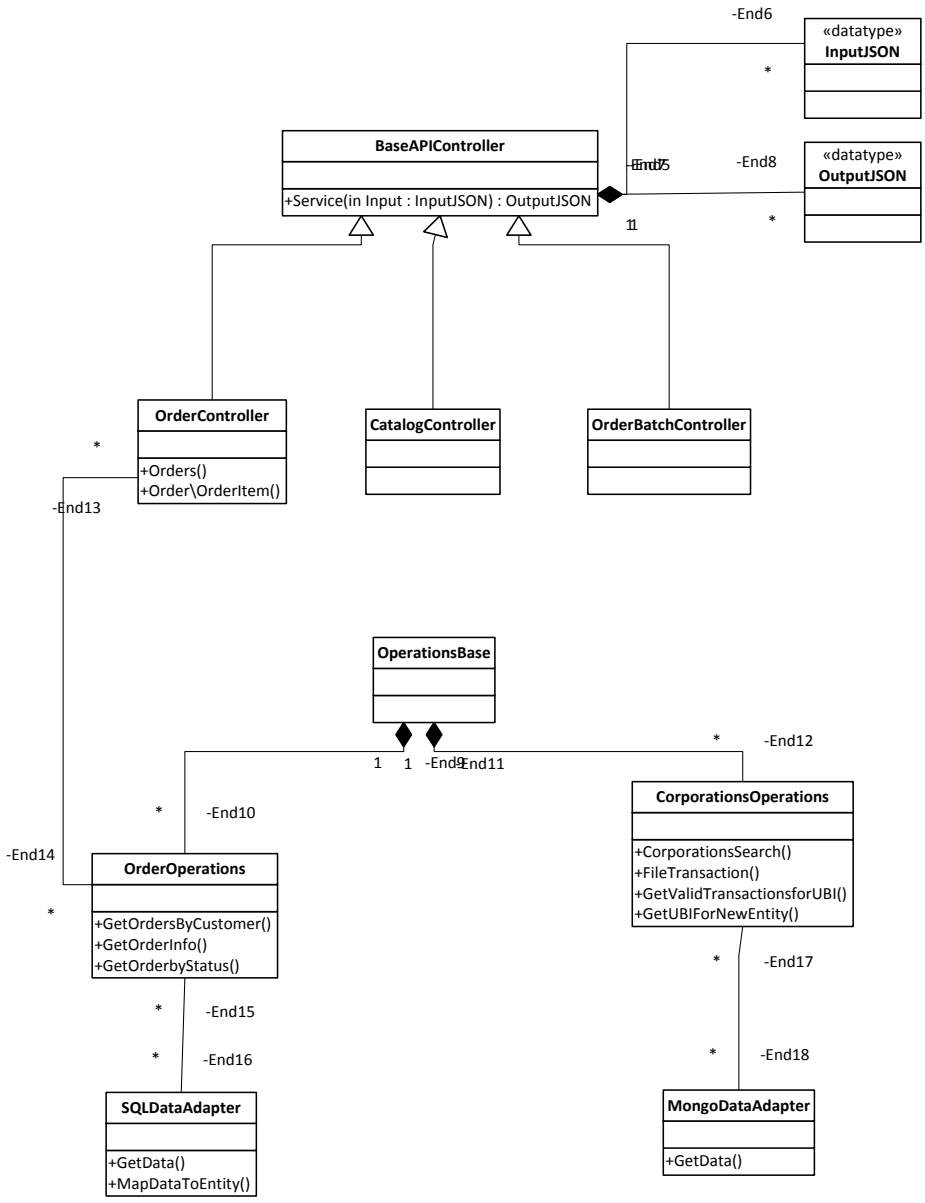
The Security Token Service will be used to issue and implement the Token. The Token will be implemented in a way that it can be seamlessly integrated into the WebAPI controller layer and the user context is available to WebAPI via the User.Identity interface built into .Net.

```
Public Class OrdersController
    Inherits ApiController
    <HttpPost(), JsonWebTokenAuthorization(SecurityRoles.SOS_STAFF,
    SecurityRoles.EBPS_SUPERVISOR, SecurityRoles.CLAIMS_UPDATE,
    SecurityRoles.MAINT)>
    Public Async Function CreateOrder(<FromBody> OrderDto As OrderDto) As Task(Of
    Object)

    //function Code to validate and create order
    OrderDTO.UpdatedBy = User.Identity.Name
    Dim securityMessage As String = String.Empty
    If Not SecurityPolicy.IsAuthorized(CType(User, ClaimsPrincipal), orderDTO,
    securityMessage) Then
         Return JSend.Fail(New With {.Message = securityMessage})
     End If
    //Do the Order Business Logic
    End function
End Class
```

The diagram below shows a base class Model for the WebAPI based services layer. The class model shows the pattern and the structure of services. Each of the services described in the REST Services Section and additional services discovered during the iterative implementation will be handled via a hierarchy of controllers, business domain objects and data objects.

Transaction data objects will be defined via a JSON data schema layer along with JSON based document storage into Mongo DB.



3. REST SERVICES

Corporations and Charities Rest Services are categories into following different categories. The URIs for these services are provided as a reference and will be determined based on the actual implementation and other SOS guidelines.

Order Services: The order services map to order domain data entities and allow the external shopping cart and internal processes to be validated and mapped to the data. Order Services serve the order pipeline receive, augment, fulfill and file and also provide mechanisms for catalog and product queries.

- **Catalog Services**: Catalog services provide the services to query the products and services received or fulfilled by the corporations and charities division plus any other divisions within SOS. The catalog is organized into categories which can have sub categories or services / products as the leaf node. The goal for the system is to limit the depth of categories to three levels but the system data model and service model does not constrain with those limits.

  A given service in the catalog can be associated with the corporations and charities transaction in the system that has Meta data and other rules associated with the transaction. The catalog system needs to allow the users to query the Meta data required for a certain transaction and any rules associated with that particular transaction.

  Transaction meta data will be defined using the JSON schema model and further extended to include any angular specific requirements along with the any business specific design patterns as they relate to corporations and charities transactions.

catalog/services

Method	Get	Comments
Description	Gets a list of sub categories or services for a given category.	A given category can contain either sub categories or services. The service can be related to a Corporations Entity Transaction or other services / products offered by SOS online.
Path Parameters	None	
Query Parameters	CategoryName: CategoryID / CatalogNodeID: Catalog Node Id	Category Name for a top level category e.g. Corporations, Charity. If no category name is provided the system returns the top level categories. CategoryID or CatalogNodeID needs to be provided for navigating the categories or services other than the top level categories.
Headers		
Recieves JSON		

Returns JSON		{
		status : "success",
		data :
		{
		"CatalogNode": {
		"Name":"Corporations",
		"ID":"233",
		"Children": [
		{"Type":"Category", "Name":"SubCategory", "ID":"233"},
		{"Type":"Product", "Name":"Annual Filing", "ID":"233", "Price":"25.00", "Taxable":"0", "MultipleQtyAllowed":"0", "Description":"Item Description"
		"EntityType":"Corporation", "SubEntityType":"LLC", "Jurisdiction":"WA","TaxElectionType":"Profit", "EntityTransactionTypeID":"23" },
		{"Type":"Product", "Name":"Initial Filing", "ID":"233", "Price":"25.00", "Taxable":"0", "MultipleQtyAllowed":"0", "Description":"Item Description"
		"EntityType":"Corporation", "SubEntityType":"LLC", "Jurisdiction":"WA","TaxElectionType":"Profit", "EntityTransactionTypeID":"23" }
		]
		}
		}
		}

catalog/services/{CatalogNodeId}/transactionMetaData

Method	Get	Comments
Description	Get more information about the service and the transaction Meta data required.	.The meta data for the catalog service for the given ID returned. This method returns a failure / Error in the JSend return code if the catalogID is a non-service catalogNode ID E.g. catalog/services/23/transactionMetaData
Path Parameters	CatalogNodeID	ServiceID / CatalogNodeID for the service required.
Query Parameters	None	
Headers		
Recieves JSON		
Returns JSON		The JSON returned by this method is the meta data for the specified filing type. The Transaction meta data JSON Schema needs to be designed to encompass the business rules and validations needed for the transaction / forms engine needed by the angular front end.

catalog/services/{CatalogNodeId}/price

Method	Get	Comments
Description	Get more detailed price information about the particular service including any old prices based on the date range.	.The price for a catalog Node ID is based on the start and end dates and can change at different times. The price used for a transaction is based on the effective date for the transaction. catalog/services/233/price
Path Parameters	CatalogNodeID	ServiceID / CatalogNodeID for the service required.
Query Parameters	startDate endDate	
Headers		
Recieves JSON		

Returns JSON		{
		status : "success",
		data :
		{
		"CatalogNode": {
		"Name":"Corporations",
		"ID":"233",
		"price": [
		{"startDate":"1/1/2009", "EndDate":"12/31/2009", "price":"75.00", "bundeledPrice":"0.0"},
		{"startDate":"1/1/2010", "EndDate":"12/31/2015", "price":"95.00", "bundeledPrice":"0.0"}
		]
		}
		}
		}

- **Orders Services**: Order services provide the shopping cart or equivalent functionality as in an eCommerce system. Even though SOS system is not a full eCommerce system there are many common traits in the SOS shopping cart vs. an eCommerce Shopping cart.
    - User can query list of their orders by getting their orders in their customerID and a status.
    - User can create a new Order and get a system assigned shopping cart ID / Order ID
    - User can add items to the shopping cart via Orders/{CartID}/Items
    - User can update the cart items and update the item meta data
    - User can validate the item or the order via post on Order via OrderActions
    - User can submit an Order along with the payment and get a confirmation back
    - Internal Users can submit orders with exceptions and also manage payments for the order along with other attributes of the payment.
    - Internal Users can manage payments
    - Internal Users can manage Order Batches and Open a Batch or Close a Batch.

/orders

Method	Get	Comments
Description	Gets a list of orders by the criteria given	Returns the basic order information on the open orders or shopping carts for a customer account. A customer can have several different shopping carts or orders that are open at a given time.
Path Parameters	None	
Query Parameters	Status CustomerID OrderBatchID	Default status is open orders only if no status is provided. The order search is one of the core services of the system and the parameters for search will be defined based on detailed mapping of requirements across several screens. /Orders?Status=”Pending” /Orders?BatchID=23
Headers	SAW Guid UserAuthToken	
Recieves JSON		
Returns JSON		Returns order Information - Shopping CartID, Order Date, Order Status, Number of Services, Order Total

/orders

Method	Put	Comments
Description	Creates a New Order for the given customer or the Internal user	Creates a new order for a given customer and returns a shopping cart ID. If the user is a web user, the order is attached the customer. If it's an internal user the order may be created as part of the batch and the user is still attributed as the creator of the order but not the customer.
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	
Receives JSON		If the user is an internal user the order may have a batch ID under which it's being created. {OrderBatchID, RecieveDate, Status, UserLocation/WorkStationID}
Returns JSON		Returns Shopping CartID as HTTP response header.

/orders/{CartID/OrderID}

Method	Get	Comments
Description	Gets Information about the Order	Returns the Order / Cart Information for a given Order. The system returns an exception if no Cart is found with the OrderID
Path Parameters	OrderID	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	System ensures that the order belongs to the Web Customer who placed the order. If the customer is an internal customer he can access the orders based on roles and permissions.
Receives JSON		
Returns JSON		Returns the Order JSON Data, Item JSON Data and Payment JSON Data (For internal users only), Also returns any order actions and status of each order Item

/orders/{CartID/OrderID}

Method	Post	Comments
Description	Updates the Information about the Order	This item provides ability to execute certain actions on the given Order.
Path Parameters	OrderID	
Query Parameters	Action	Validate – Validates the Order Submit - Submits the Order from the external Web / Shopping cart. The Submit also requires the payment information and ensures that the payment is matching the items and that the card can be processed.
Headers	SAW Guid UserAuthToken	System ensures that the order belongs to the Web Customer who placed the order. If the customer is an internal customer he can access the orders based on roles and permissions.
Receives JSON		Order Information, Payment Information
Returns JSON		

/orders/{CartID/OrderID}

Method	Delete	Comments
Description	Deletes the Order	Deletes the given Order from the system. System returns an exception if the given Order is in a status where it can't be deleted.
Path Parameters	OrderID	
Query Parameters		-
Headers	SAW Guid UserAuthToken	System ensures that the order belongs to the Web Customer who placed the order. If the customer is an internal customer he can access the orders based on roles and permissions.
Receives JSON		
Returns JSON		

/orders/{OrderID}/Items/{Number}/TransactionData

Method	Get	Comments
Description	Gets the order transaction data for a given order ID and a given Item	The transaction data is the data associated with the order Item Transaction. The transaction data will be validated and saved as JSON data. The data for the transaction will be validated against the JSON Schema and other rules for transaction. The get function returns the saved transaction data along with any validation errors.
Path Parameters	OrderID Number	Order ID of the Order Number of Order Item in a given Order
Query Parameters	None	
Headers	SAW Guid UserAuthToken	
Receives JSON		
Returns JSON		Returns the transaction JSON data along with any transaction validation errors or validation status.

/orders/{OrderID}/Items

Method	Put	Comments
Description	Creates a New Order Item under the given order	This method creates an Item in a given order. If there is transaction data associated with the item, transaction data is validated by the server and error is returned.
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	
Receives JSON		{ “CatalogNodeID”:”233”, “EntityId”:”3345”, “Price”:”75.00”, “TransactionData”:”##################”}
Returns JSON		Returns the Order Item URI.

/orders/{OrderID}/Items/{number}

Method	Post	Comments
Description	Updates a given item in the shopping cart.	This is the way for the user to update the item or the item transaction data updated in a given order. Most of the time this is how the user will update the transaction data for a given shopping cart item.
Path Parameters	None	
Query Parameters	Action	Add – Adds the Item to the Order, Validates the serviceID / Catalog ID is valid catalogId. Save – Saves the Item without Validation, the validation rules are not enforced on save Validate – Validates the current Item and Transaction data using the business rules for the transaction Complete – Marks the Item Completed, Validates the Item and saves the transaction data CheckOut – Checks out the given Order Item for working by an internal user CheckIn – CheckIn a specific order item and its related transaction data. Receive – Save the Order Item with exceptions RecieveWithExceptions – Marks the Item as received but with Errors
Headers	SAW Guid UserAuthToken	
Receives JSON		{ “CatalogNodeID”:”233”, “EntityId”:”3345”, “Price”:”75.00”, “TransactionData”:”##################”}
Returns JSON		Returns the Order Item URI.

/orders/{OrderID}/Items/{number}

Method	Delete	Comments
Description	Deletes the OrderItem from the Order	This removes the Order Item from the Order / Shopping cart.
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	The headers are used to ensure that the order belongs to the user performing the operation on the order.
Receives JSON		
Returns JSON		

/orders/{OrderID}/Payments

Method	Get	Comments
Description	Returns the payments for the order	This method only returns data for internal user roles only
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	System ensures that the user is in the internal user role.
Receives JSON		
Returns JSON		Returns a list of payments for the given order.

/orders/{OrderID}/Payments

Method	Put	Comments
Description	Creates a New Order Payment under the existing order	The payment method for separately adding the payment information to the order is only available to internal users. External users will call the Submit Order and submit the payment along with the Order Submission.
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	System ensures that the user is in the internal user role.
Receives JSON		Payment Information
Returns JSON		Returns the new Payment URI

/orders/{OrderID}/Payments/{PaymentID}

Method	Post	Comments
Description	Updates the Payment based on the action.	Updates the given Payment under a given order with different actions requested. A payment may be posted but payment of different types e.g. cards can have different actions.
Path Parameters	None	
Query Parameters	Action	Update Cancel Commit.
Headers	SAW Guid UserAuthToken	System ensures that the user is in the internal user role. External users provide the payment information using the order Submit action.
Receives JSON		Payment Information and action.
Returns JSON		

- **Register**: Register / Till Services allow the receiving team to receive the Orders, create order batches and perform other operations or the orders and order batches from the staff console.
    - Internal User can create a new Batch
    - Internal Users can manage Order Batches and Open a Batch or Close a Batch.
    - Internal Users can add orders to the batch – The API for Orders under a batch will be very similar but, just the fact that Order will belong to the batch. The orders belonging to the batch won't be released for further processing until the batch is committed. Order and Item API will function under a given batch or without a batch.

/orderBatches

Method	Get	Comments
Description	Gets a list of order batches by the criteria given	Returns the basic order batch information
Path Parameters	None	
Query Parameters	OrderBatchID Date BatchUserID Status	Other batch search criteria can be used to search a particular order batch.
Headers	SAW Guid UserAuthToken	Order Batches can only be created by Internal users so far. If there are external partners who will be allowed to leverage order batch API it will need to be evaluated on how to secure and allows a user Id for the partner accounts.
Recieves JSON		
Returns JSON		Returns order Batch Information

/orderBatches

Method	Post	Comments
Description	Creates a New Order batch for the given internal user	Creates a new order batch for the internal user, any information for the user. Desk / register and other session data are captured.
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	
Receives JSON		Batch creation data .. register, date, user, etc.
Returns JSON		Returns Shopping OrderBatchId as HTTP response header.

/orderBatches/{OrderBatchID}

Method	Post	Comments
Description	Updates the given Order Batch	Updates the given Order Batch using the actions on the order batch
Path Parameters	None	
Query Parameters	Action	Close – Closes the given batch. The batch is closed to create reconciliation and other data ReOpen – An closed batch can be reopened if issues are found in reconciliation Commit – This commits the batch and posts the transactions to the Revenue.
Headers	SAW Guid UserAuthToken	
Receives JSON		Batch creation data .. register, date, user, etc.
Returns JSON		

/orderBatches/{OrderBatchID}

Method	Get	Comments
Description	Returns the Batch Summary for the given Order Batch	Returns the Batch Summary data that can be used to display the batch information as well as for reconciliation
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	
Receives JSON		
Returns JSON		Returns Batch Summary data, batch status – Order summary data as well as payment summary data needed for reconcillation.

/orderBatches/{OrderBatchID}

Method	Delete	Comments
Description	Deletes the given OrderBatch	Deletes the given order batch if it is in a state where the batch can be deleted. Once Payments and other info is posted the order batches cannot be deleted.
Path Parameters	None	
Query Parameters	None	
Headers	SAW Guid UserAuthToken	
Receives JSON		
Returns JSON		

- **Work Queues**: Work / Order Queues services allow the SOS internal staff in various business units to handle the order pipeline. Most of the Work Queue API's act on the order and the Item information
    - Internal User can Query Orders – This can be fulfilled by the Orders Service using the search criteria. The search allows to get a certain types of orders and order items to build a specific work queue.
    - Internal User can Check out an OrderItem to work on. CheckOut is done on the OrderItem via the checkout action.
    - CheckIn – CheckIn can be done on the Order Item the user is working on and can mark the item explicitly completed or checkIn the item.
    - Internal users can take other actions on the order Item as needed.
    - Internal Users can file the Transaction related to the orderItem by calling this filing API or using the complete action of the OrderItem API.

Filing Services: The filing services map to the filing services map to Filing domain data entities and allow the validation of filing information, filing rules, filing documents and data integrity of the filed documents. Filing services are core services in the system with a set of rules that are driven filing Meta data and also set of custom rules where they are required. Filing Services also generate the actual filed documents for electronic transactions for record management and audit purpose.

Filing/Entities

Method	Get	Comments
Description	Gets a list of entities that match the given criteria	An entity can be of different type and all entities of different types are stored in the NOSQL data base. Each entity may have different attributes based on the entity type and the attributes will be returned accordingly.
Path Parameters	None	
Query Parameters	EntityName: EntityType EntitySubType Jurisdiction ProfitStatus EntityStatus	Various Search Criteria for Entities. Entities can be search by attributes, partial names, governing officer name and other criteria. Entities could be queried for status and other aspects to create dissolutions and other things.
Headers		
Recieves JSON		

Returns JSON		{
		status : "success",
		data :
		{
		"totalResults":2,
		"startCount":1
		"results": [
		{"EntityTypeType":"Corporation", "SubEntityType":"LLC"", "Jurisdiction":"WA","TaxElectionType":"Profit", "LegalNameName":"John and Doe LLC",
		"BusinessAddress": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}
		"RegisteredAgent":{"Type":"Entity","UBI":"23334445","Address": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}}
		},
		{"EntityTypeType":"Charity", "SubEntityType":"", "Jurisdiction":"WA","TaxElectionType":"NonProfit", "LegalNameName":"John and Doe LLC",
		"BusinessAddress": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}
		"RegisteredAgent":{"Type":"Entity","UBI":"23334445","Address": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}}
		},
		]
		}
		}

Filing/Entities/EntityInfo/

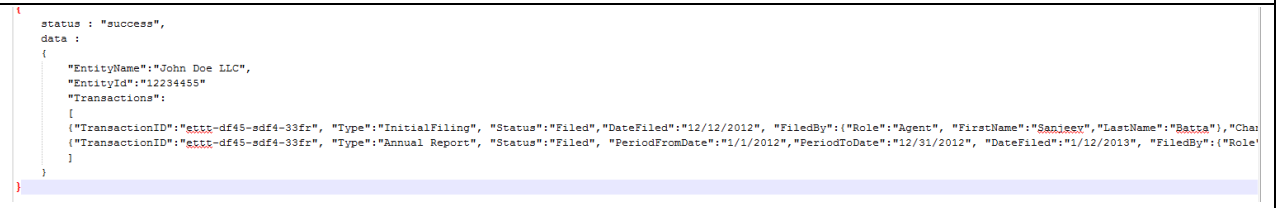
Method	Get	Comments
Description	Gets the detailed information about the entity	The method returns the basic information about the entity. Each Entity is of different type and returns the JSON data that matches that entity type. JSON data allows the flexibility to return different types of entity information in consistent but flexible fashion.
Path Parameters		
Query Parameters	UBI or EntityID or CharityId	Various Identifiers by which the entity can be uniquely identified.
Headers		
Recieves JSON		

Returns JSON		{
		status : "success",
		data :
		{
		"CharityInfo":
		{
		"EntityTypeType":"Charity", "SubEntityType":"", "Jurisdiction":"WA","TaxElectionType":"NonProfit", "LegalNameName":"John and Doe LLC",
		"BusinessAddress": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}
		"RegisteredAgent":{"Type":"Entity","UBI":"23334445","Address": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}}
		"SoliciationMethods": ["Internet","DoorToDoor", "Radio","TeleMarketing", "Other"],
		"PersonResponsibleForFinancials": {"FirstName":"Robert","LastName":"Johnson","Address": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"W
		"PersonAcceptingResponsibility": {"FirstName":"Robert","LastName":"Johnson","Address": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA
		"CharityFinance": ["FiscalBeginDate":"1/1/2013","FiscalEnddate":"12/31/2013", "BeginingAssets":"2,000,000","TotalExpenses":"1,900,000", "ProgramServices":"200,000"],
		"CharityOfficers":[
		{"Title":"President","FirstName":"Robert","LastName":"Johnson","Address": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}}
		{"Title":"Secretary","FirstName":"Robert","LastName":"Johnson","Address": {"line1":"1234 Vista Lane","line2":"Suite 2002","line3":"", "city":"Olympia", "State":"WA"}}
		}
		}
		}

Filing/Entities/EntityInfo/{EntityID}/AllowedTransactions

Method	Get	Comments
Description	Gets the Transactions for the Entity	This method returns the Allowed Transactions for a given EntityID. The Allowed Transactions are based on what current filings have been already done on this entity and what are pending or allowed transactions in the current state of this entity.
Path Parameters	EntityID	
Query Parameters		
Headers		
Recieves JSON		
Returns JSON		Returns allowed transactions and their due dates for the purpose of validating and only allowing users to file those transactions. This function may include the calculation and other logic for reinstatement and show the overall price for each transaction. In that case it could be something across the catalog and the corporations boundary.

Filing/Entities/EntityInfo/{EntityID}/Transactions

Method	Get	Comments
Description	Gets the Transactions for the Entity	This method returns the different transactions that have occurred on that entity given an EntityID
Path Parameters	EntityID	
Query Parameters	IncludeDeleted IncludePending	Internal users can get Pending Transactions as well as any deleted Transactions. The system will validate that those transactions are returned for internal users only. The status of each transaction is also returned.
Headers		
Recieves JSON		
Returns JSON		

Filing/Entities/EntityInfo/{EntityID}/Transactions

Method	Post	Comments
Description	Posts a Transaction for this Entity	This is the only method to make changes to an entity record. A transaction record is posted for all changes. The transaction data is validated against the meta data and transaction rules along with the state of entity. Transaction data is then posted to the entity in a way that any information required at the entity level is updated. A document / PDF version for the Transaction is also created and stored along with the Transaction for record management reasons.
Path Parameters	EntityID	
Query Parameters	Action	Validate – Validates the transaction without posting the transaction to the entity. Both the meta data level validation as well as transaction level validation is done for the entity integrity Post – Posts the Transaction to the entity and updates and creates the required audit trail
Headers		
Recieves JSON		Different Transaction data is posted based on a number of different transactions. Transaction data JSON for each transaction will need to be defined at the development stage. An example of such Transaction JSON is provided.
Returns JSON		

Filing/Entities/EntityInfo/{EntityID}/Transactions/{TransactionID}

Method	Get	Comments
Description	Returns the Transaction Data	Returns the data about a Transaction given a transaction ID
Path Parameters	EntityID TransactionID	
Query Parameters		
Headers		
Recieves JSON		
Returns JSON		Different Transaction data is returned based on the transaction type. Transaction data JSON for each transaction will need to be defined at the development stage. An example of such Transaction JSON is provided. Any filed Documents for the Transaction are also returned.

Filing/Entities/EntityInfo/{EntityID}/Transactions/{TransactionID}

Method	Post	Comments
Description	Returns the Transaction Data	Allows Internal users or system process to update a given transactions data. The update should be allowed in a very restrictive fashion to allow the system to maintain integrity.
Path Parameters	EntityID TransactionID	
Query Parameters	Action	Reverse – Reverses the transaction. Any updates made via this transaction along with the filing are removed. The transaction is marked as deleted.
Headers		
Recieves JSON		Status, Other attributes in a Transaction
Returns JSON		

Filing/Entities/EntityInfo/{EntityID}/Transactions/{TransactionID}/FiledDocuments/{Num}

Method	Get	Comments
Description	Returns the PDF version of any documents filed	Returns the PDF version of the filed documents for the given transaction and sequence number.
Path Parameters	EntityID TransactionID	
Query Parameters		
Headers		
Recieves JSON		
Returns JSON		A Document PDF is returned and is not in the JSON format but as a binary stream encoded as Document/PDF

Filing/Entities/EntityInfo/{EntityID}/Transactions/{TransactionID}/FiledDocuments

Method	Put	Comments
Description	Add a document to the transaction	Add a document to the transaction and returns a document ID for the added document
Path Parameters	EntityID TransactionID	
Query Parameters		
Headers		
Recieves JSON		Accepts a binary stream of document. This is not in JSON format.
Returns JSON		Returns a document ID / Document URL for the uploaded document.

Filing/NameSearch

Method	Get	Comments
Description	Returns if the name is a valid name given Namesearch for a given entity type, sub entitytype	The name search rules are applied based on the entity Type and Entity sub type and name if available is returned along with possible names.
Path Parameters		
Query Parameters	EntityType SubEntityType SuggestedName	
Headers		
Recieves JSON		

Returns JSON		{
		status : "success",
		data :
		{
		"EntityType":"Corporation",
		"EntitySubType":"LLC"
		"RequestedName":"John Doe",
		"MatchingNames":
		[
		{"SuggestedName":"John Doe LLC","Available" = "False"},
		{"SuggestedName":"John Doe Lunch bags LLC","Available" = "True"}
		]
		}
		}

Filing/Entities/EntityInfo/{EntityID}/Notices

Method	Get	Comments
Description	Get a list of notices for the given entity	Returns a list of notices for a given entity
Path Parameters	EntityID	
Query Parameters		
Headers		
Recieves JSON		Accepts a binary stream of document. This is not in JSON format.
Returns JSON		Returns a document ID / Document URL for the uploaded document.

Filing/Entities/EntityInfo/{EntityID}/Notices

Method	Put	Comments
Description	Record a Notice for the entity .	Record the notice for a given entity. Adds the new notice to the entity record.
Path Parameters	EntityId	
Query Parameters		
Headers		
Recieves JSON		Accepts the Notice JSON Data (Date, Type, Notes) as well as the binary document stream.
Returns JSON		

User and Notification Services : The user and notification services allows getting the information on users and notifications and enables the different notifications to be managed and delivered. Notifications are delivered when certain events happen on an entity or an order.

Users

Method	Get	Comments
Description	Get a User Info record	Get the Information for a given user. The data returned may vary based on the permissions of the callers UserID. If the external caller calls the service it may only retrieve its own user information.
Path Parameters	None	
Query Parameters	SAWGUIID: SOSADUserId	
Headers	UserID	
Recieves JSON		
Returns JSON		User Information data is returned.

Affiliate\{AffiliateID}\Users

Method	Get	Comments
Description	Gets a list of users for the given affiliate or an expert company Id	This returns the list of user for a given affiliate ID. If the external user is calling this service the security will be validated for him to be in the appropriate role for the affiliate to get the list of users for the given affiliate.
Path Parameters	None	
Query Parameters	AffiliateID	
Headers	UserID	
Recieves JSON		
Returns JSON		List of users for the given affiliate is returned.

Affiliate\{AffiliateID}\Users

Method	Put	Comments
Description	Add a user to a given affiliate ID record.	This service adds the given user record to an affiliate entity. The user calling the service to add other users must be an affiliate administrator.
Path Parameters	None	
Query Parameters	AffiliateID	
Headers	UserID	
Recieves JSON		User Info JSON data.
Returns JSON		Success or failure

Affiliate\{AffiliateID}\Users\{UserID}

Method	Delete	Comments
Description	Remove a given user from the affiliate relationship.	This service removes the given user record to an affiliate entity. The user calling the service to add other users must be an affiliate administrator.
Path Parameters	AffiliateID UserId	
Query Parameters	AffiliateID	
Headers	UserID	
Recieves JSON		User Info JSON data.
Returns JSON		

Users\{UserID}\Subscriptions

Method	Get	Comments
Description	Gets the List of Subscriptions for the user	Returns a list of different subscriptions for this UserID.
Path Parameters	None	
Query Parameters	SAWGUIID: SOSADUserId	
Headers	UserID	
Recieves JSON		
Returns JSON		Subscription information containing - SubscriptionID, Date, Type, EntityId, OrderID, TransactionTypes is returned.

Users\{UserID}\Subscriptions

Method	Put	Comments
Description	Creates a new subscription for the user.	Create a new Subscription for a given entity by EntityID or OrderID
Path Parameters	UserID	
Query Parameters	SAWGUIID: SOSADUserId	
Headers	UserID	
Recieves JSON		Subscription information containing - SubscriptionID, Date, Type, EntityId, OrderID, TransactionTypes is posted. A subscription is created with the information and attached to the entity.
Returns JSON		

Users\{UserID}\Subscriptions\{SubscriptionID}

Method	Delete	Comments
Description	Remove the subscription given a subscriptionId	Deletes the subscription from the entity, order and user subscriptions. System ensures that the Subscription ID belongs to the user requesting the deletion.
Path Parameters	UserID	
Query Parameters	SAWGUIID: SOSADUserId	
Headers	UserID	
Recieves JSON		
Returns JSON		

4. INTEGRATION DESIGN

Revenue Integration: The integration between the SOS revenue system and the new Corporations and Charities system needs to be asynchronous and be loosely coupled in a way that it does not depend on the revenue system directly. The Order system can receive orders, payments and apply those payments to the order items. A catalog system supports the Order system and the mapping of the item price to different revenue classes.

In case of orders processed via the mail or front counter the order revenue is posted to revenue at the completed commit of an Order batch. Once a batch is committed the Order and its revenue is mapped and posted to the Revenue system. In case of online orders the order is posted to revenue at the final posting of the transaction to the filing system.

The mapping of the Order system fields to Revenue System fields is as follows:-

- Order.RevenueTrackingID – tblTracking.TrackingID
- OrderItem. RevenueDocumentID – tblDocument.DocumentId
- Payment. RevenueReceiptID – tblReceipt.ReceiptKey
- OrderItemPaymentRevenueSplit. RevenueSourceKey - tblSource.SourceKey

Any reversal to the order posts similar reverse entries in the sourcing and receipt tables. Any adjustment to the order produces the equivalent balancing sourcing reversals and then post the right transaction and source amounts to the revenue system.

The overall design and mapping approach with the revenue integration will be based on either a contract first stored procedure call or Web services interface over the existing revenue system. Order system and the revenue system will share the catalog, products and pricing information.

Project Approval
Signatures

The signatures below indicate the Logical architecture model was reviewed by all parties and approve of its content.

_____

SOS Technical Steering Committee