1 Overview

The Service NSW Omni Channel Reference Architecture (OCRA) provides a pattern for the delivery of capability into the Service NSW Ecosystem.

The overall aim of the architecture pattern is to provide omni channel delivery of capability. That is where capability delivered in one program or channel, can be leveraged and provided through alternate channels without the need for major refactoring or rework.

1.1 Constraints

The OCRA works within a number of pre-defined constraints:

- AWS
- PCF
- Apigee

2 Value Added Services

2.1 Overview

Value Added Service (VAS) provide abstracted common capability available to any internal Transaction Logic Component.

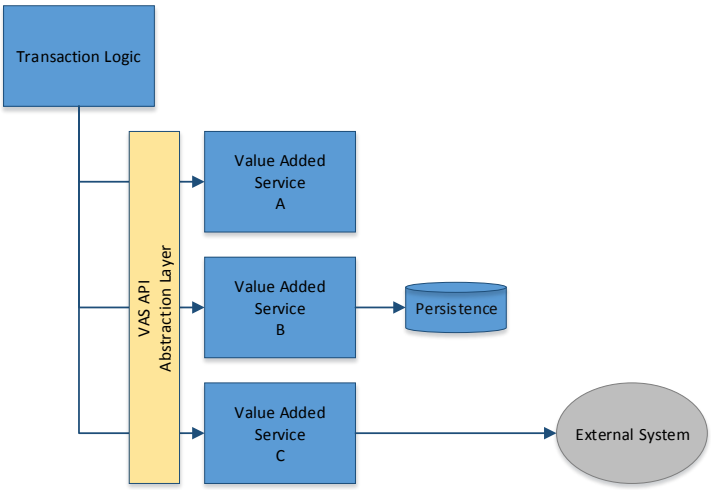
The VAS are accessed via an abstraction layer.

The VAS will provide an atomic level of functionality (single targeted capability).

The VAS will have no knowledge of which Transaction Component is calling them (outside of passed data).

The VAS may internally implement capability such as Persistence, or External System integration to assist in completing its function.

The intent is for the number of VAS to grow over time, that when implementing Transaction Logic, if a capability is required that can be used across transactions, then this capability should be developed independently as a VAS.



2.2 Catalogue of Services

A catalogue of VAS is required. To date no such catalogue has been produced.

It is anticipated that the number of VAS required in the ecosystem is quite small (at around 20-30 components).

The following VAS have been identified:

- In Production
  - Transaction Logging
  - Metrics Logging
  - Receipt Generation
  - SMS
  - Email
  - Case Management

  - Customer Management
  - Payment Management
  - JWT Management
  - Transaction Configuration
  - Proof of Identity
- Under Development
  - File Writing
  - File Reading
  - Error Mapping
  - Save / Resume
  - Notifications
  - Service Management
- Future Target Capability
  - Encryption / Decryption
  - Caching
  - Image Store
  - Signature Store
  - Document Store

2.3 VAS

2.3.1 Transaction Logging

The log is “Write Only”, where data can only be added to the log, never updated, modified or deleted.

In the initial deployment the logs are write only, with no capability for read. This protects any data held within the log as there are no enabled “read” functions to access the data.

In future releases data held within the log file will need to be encrypted. Particularly if transactions are logging sensitive or personal information. In this situation the core metadata may remain unencrypted, allowing access, sorting, and analysis, while the data can be encrypted, thus protecting it against unauthorised access.

In the Transaction Log where the log is required to store a document as part of the log, the document is split from the log and stored in a separate table. Hence one call to the Tx Log may result in two different table entries.

The mechanism used for storage of the log data itself is a MS SQL Server DB held in AWS.

2.3.2 Metrics Logging

Metrics Log – is only associated with the timing and execution of transactions, and should have recorded in it the start and end time of a transaction through any key component of the system.

The log is “Write Only”, where data can only be added to the log, never updated, modified or deleted.

In the initial deployment the logs are write only, with no capability for read. This protects any data held within the log as there are no enabled “read” functions to access the data.

In future releases data held within the log file will need to be encrypted. Particularly if transactions are logging sensitive or personal information. In this situation the core metadata may remain

unencrypted, allowing access, sorting, and analysis, while the data can be encrypted, thus protecting it against unauthorised access.

The mechanism used for storage of the log data itself is a MS SQL Server DB held in AWS.

2.3.3 Receipt Generation

The receipt generator takes input from the calling transaction and converts this input into a Service NSW PDF based receipt (using SNSW letterhead and copy etc).

The result of the process is that a PDF file is stored in local storage against the Customer and Transaction, and returned to the calling transaction.

Where a copy of a receipt is requested, a previously stored receipt is retrieved and returned to the calling transaction.

The mechanism used for storage of the receipt data itself is through a MS SQL Server DB held in AWS.

As the receipt may record sensitive and personal information the data held in the receipt data store may require encryption.

2.3.4 SMS

The SMS Notification component allows the calling transaction to send SMSs to the Customer.

The mechanism is based on sending the SMS to a known Mobile Number.

SMSs are sent via the existing Twillio Gateway.

2.3.5 Email

The Email Notification component allows the calling transaction to send emails to the Customer.

The mechanism is based on sending the email to a known email address.

Emails are sent via the existing SendGrid Gateway.

2.3.6 Notification

This component allows the sending of Messages to the Customer based on the Customers existing profile.

The component should be passed the Message, Transaction Type, and the Customer Id, where it should lookup the Customer within Service NSW, determine the Customers Messaging Preference (Email, SMS, Both, None) for the given Transaction Type, and send the message using that preference.

2.3.7 Encryption / Decryption

The encryption capability allows the encryption and decryption of data stored in secure data stores. The services offered are:

- Encrypt Data
- Decrypt Data

The algorithm used for Encryption/Decryption will be specified by the security team.

The component does not store the encrypted data, it is up to the calling transaction to store the encrypted/decrypted result if needed.

2.3.8 Case Management

The Case Component allows the maintenance (search, read, creation, update, delete etc) of Case Data.

Case Data is the domain of SalesForce, however management and display of this information can occur within any transaction, or dashboard.

2.3.9 Customer Management

The Customer Component allows the maintenance (search, read, creation, update, and delete/terminate) of customer data.

There are many sources of Customer Data, which can generally be partially sourced from SNSW directly, or from each of the target agencies. This component allows the retrieval (and management) of Customer Data both known to SNSW and a specific agency in a uniform way.

2.3.10 Payment Management

The Payments Component allows the collection of CC and Cash payments over the counter (or digitally) through the OneGov GLS Payment System.

The Component is required to support Payments, Refunds, and Reconciliation.

On completion of a payment the component should redirect the user to complete the transaction.

It is envisaged that the payment component will utilise the existing PinPads (and Cash Draws) on the CSR Work Station, and not require the manual entry of CC Details.

2.3.11 JWT Management

The generation of JWT Security Tokens should be handled centrally to ensure that all transactions utilise the same approach for Security Management.

The component should allow the generation and retrieval of JWTs.

2.3.12 Transaction Configuration

The Transaction Configuration component allows the retrieval of Transaction Setup data (such as Drop Down List values).

This information can be sourced from different locations depending on the transactions requirements.

For Service NSW related data, this information is sourced from a temporary data store.

For Agency related data, this information is sourced from the downstream agency.

On calling this capability the caller must supply the transaction type to allow the identification of the correct setup data.

2.3.13 File Writer

Allows the writing of a file (Text, CSV, XML, JSON, Doc, PDF etc) to a defined location (S3 bucket).

2.3.14 File Reader

Allows the reading of a file (Text, CSV, JSON, XML, PDF, Doc etc), from a defined file location (such as an S3 bucket).

2.3.15 Caching

Caching allows a transaction to take advantage of a temporary storage space (a cache).

- Data can be pushed into the Cache.
- Data can be retrieved from the Cache.
- Data is segregated within the Cache based on Transaction Type.

Note: the cache will be provisioned through temporary storage, and so can go unavailable without notice.

Note: the cache is not intended to support wider "Search" or "Update" capability, simply write/retrieve based on a Transaction Type and Id.

2.3.16 Error Mapping

The Error Mapping component maps a provided (Agency) error message to a Customer Centric Error Message. It relies on a defined set of error message maps which is stored in a temporary database.

When the component receives a request the calling Transaction Logic component should provide the Error Message, Error Id, and Source System/Transaction. If an existing Map for this error message is found in the Mapping database then the associated error message is returned. If no map is found in the Mapping database then a default message is returned (the equivalent of “Oops something went wrong please try again later”, and a new entry is added to the database for later remediation).

2.3.17 Image Store

The image store contains encrypted copies of the Customer Photo.

The intent is that this image is reusable across all customer interactions and transactions where a photo is required.

2.3.18 Signature Store

The signature store contains encrypted copies of the Customer Signature.

The intent is that this signature is reusable across all customer interactions and transactions where a signature is required.

2.3.19 Document Store

The Document Store contains encrypted copies of Customer Documents.

The intent is that these documents are reusable across all customer interactions and transactions where a document is required.

The content will require encryption which is keyed to the Customers identity (ie all documents encrypted differently for each customer).

2.3.20 Save / Resume

Provides the capability to Save and Resume a transaction mid flight.

This capability supports the services:

- Pause Transaction
- Resume Transaction

It is intended to be used for transactions where there is a high likelihood of a “walkout” such as during a manual test (where there is a natural break in a transaction), before a payment is made, where the customer can leave and return at a later date.

The component should allow the transaction state to be written to temporary storage and retrieved at a later date. On retrieval the transaction data should be removed from the temporary storage. Equally the temporary storage should only house data for a maximum set period (suggest 5 days).

In writing the transaction to the temporary storage some core metadata is required outside that of the transaction data to allow correct identification at a later date. This includes:

- Transaction Id
- Transaction Type
- Transaction Date
- Transaction Name
- Customer Id

2.3.21 Proof of Identity

POI is the conglomeration of a proof of identity function, where a Customers Image, Signature, and Identity credentials can be stored by SNSW and used at a later date in other applications. Thus allowing applications requiring POI to be performed online, and streamlining the process for future CSR transactions.

Note this can not be done without the customers consent.

The function relies on creating secure data stores, so relies on Encryption, and storage of Image, Signature and POI information.

The POI component should store the identity level that the Customer has been certified to, these being:

- Uncertified
- Document Certified
- Visually Certified

And the storage of the Point System associated with the certification.

This data must be encrypted prior to storage, and decrypted on retrieval.