Change ID		**Issue ID**	
Requestor:			
Name & Surname	Lucky Murovhi	Date:	26 June 2023
Project Name:	High speed Interface (SOAP to REST)		
Submission Date:	26 June 2023		
Priority:	High **Medium** Low		

1. CONTEXT

The SDBSS is undertaking an initiative to migrate from SOAP to REST, embracing a modern and lightweight communication approach. This migration aims to enhance performance, simplify integration, and align with industry standards for seamless data exchange. The migration from SOAP to REST for a high-speed interface offers several advantages. Firstly, REST is inherently more lightweight compared to SOAP. It has a simpler structure and a smaller message footprint, resulting in faster transmission and reduced overhead. This is especially crucial for high-speed interfaces that require rapid data exchange.

2. PROPOSED CHANGE/REQUIREMENT

The proposed change or requirement for migrating the SOAP-based High-Speed Interface (HSI) service to REST with functions (Execute, Perform, and Query) should cater for the following development considerations:

- **Endpoint Design:** The RESTful architecture should include well-defined endpoint for each function, ensuring that the URL structure and resource naming are intuitive and aligned with the purpose of each operation. The endpoint should be designed to be easily understandable and discoverable by developers and clients consuming the API.
- **HTTP Methods:** The RESTful API should utilize appropriate HTTP methods to correspond with the desired actions. For example, the Execute and Perform functions, which involve performing specific actions, can be mapped to HTTP POST or PUT methods. The Query function, which involves retrieving data, can be associated with the HTTP GET method.
- **Request and Response Formats:** The migration should consider the request and response formats compatible with REST. JSON (JavaScript Object Notation) is commonly used in RESTful APIs due to its lightweight and widely supported nature. The development should cater for transforming the SOAP-based request and response structures to JSON format, ensuring compatibility with the RESTful architecture.

- **Authentication and Authorization:** The migration should address the authentication and authorization mechanisms required for accessing the RESTful API. It should include considerations for implementing secure authentication methods such as OAuth or token-based authentication to ensure proper access control and protection of sensitive data.
- **Error Handling:** The development should incorporate appropriate error handling mechanisms in the RESTful API to provide clear and informative error responses. This includes defining error codes, error messages, and providing guidance on handling exceptions and errors in a consistent and standardized manner.
- **Documentation and API Design Standards:** The migration should include comprehensive documentation for the RESTful API, describing the available endpoints, their expected behaviors, input parameters, and response formats. It is important to adhere to API design standards and best practices, such as using meaningful resource names, consistent naming conventions, and providing clear examples for developers to consume the API effectively.
- **Testing and Validation:** The development should cater for thorough testing and validation of the RESTful API to ensure that the migrated functions (Execute, Perform, and Query) behave as expected and meet the requirements. This includes functional testing, integration testing, and performance testing to validate the functionality and performance of the API.

During the migration process, it is crucial to ensure that the functionality provided by these SOAP functions is maintained in the RESTful architecture. Careful analysis and consideration should be given to the specific requirements and behaviors of each function, and suitable RESTful design patterns and practices should be applied to achieve the equivalent functionality in a RESTful manner.

3. SAMPLES

Endpoint: http://10.115.243.142:7003/APD/Provision

3.1 Execute Function

3.1.1 Request

Request

```
prv:ExecFunctionQuery xmlns:prv="http://xml.vodacom.com/schema/PRV"
xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
   <prv:FunctionSelector>string</prv:FunctionSelector>
   <prv:AttributeList>
      <!--1 or more repetitions:-->
      <com:Attributes>
         <com:Name>string</com:Name>
         <com:Value>string</com:Value>
      </com:Attributes>
   </prv:AttributeList>
</prv:ExecFunctionQuery>

//……………………………….Example…………………………………….

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns3:ExecFunctionQuery xmlns:ns2="http://xml.vodacom.com/Schema/APD/Common"
xmlns:ns3="http://xml.vodacom.com/schema/PRV">
```

```
      <ns3:FunctionSelector>SPNREG</ns3:FunctionSelector>
      <ns3:AttributeList>
        <ns2:Attributes>
          <ns2:Name>SUB ID</ns2:Name>
          <ns2:Value>0665110159</ns2:Value>
        </ns2:Attributes>
        <ns2:Attributes>
          <ns2:Name>SUB ID TYPE</ns2:Name>
          <ns2:Value>MSISDN</ns2:Value>
        </ns2:Attributes>
      </ns3:AttributeList>
    </ns3:ExecFunctionQuery>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.2 Response

Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <ns1:AttributeList xmlns:ns1="http://xml.vodacom.com/schema/PRV">
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>REQUEST SEQ</common:Name>
        <common:Value>1302129958</common:Value>
      </common:Attributes>
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>REQUEST ID</common:Name>
        <common:Value/>
      </common:Attributes>
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>Send SMS Config YN</common:Name>
        <common:Value>N</common:Value>
      </common:Attributes>
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>SU_ID</common:Name>
        <common:Value>147673503</common:Value>
      </common:Attributes>
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>CUSTOMER_ID</common:Name>
        <common:Value>C0022968</common:Value>
      </common:Attributes>
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>EXECUTION TP</common:Name>
        <common:Value>ASYNC_CONFIG</common:Value>
      </common:Attributes>
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>SU_SEQ</common:Name>
        <common:Value>147673503</common:Value>
      </common:Attributes>
      <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
        <common:Name>PENDING_YN</common:Name>
        <common:Value>N</common:Value>
      </common:Attributes>
    </ns1:AttributeList>
  </soap:Body>
</soapenv:Envelope>
```

3.2 Perform Function

3.2.1 Request

Request

```
<prv:PerformFunctionQuery xmlns:prv="http://xml.vodacom.com/schema/PRV"
xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
  <prv:FunctionSelector>string</prv:FunctionSelector>
  <prv:AttributeList>
    <!--1 or more repetitions:-->
    <com:Attributes>
      <com:Name>string</com:Name>
      <com:Value>string</com:Value>
    </com:Attributes>
  </prv:AttributeList>
</prv:PerformFunctionQuery>

//…………………………….Example……………………………

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns3:PerformFunctionQuery xmlns:ns2="http://xml.vodacom.com/Schema/APD/Common"
xmlns:ns3="http://xml.vodacom.com/schema/PRV">
      <ns3:FunctionSelector>QSBSUM</ns3:FunctionSelector>
      <ns3:AttributeList>
        <ns2:Attributes>
          <ns2:Name>SUB ID</ns2:Name>
          <ns2:Value>0665110159</ns2:Value>
        </ns2:Attributes>
        <ns2:Attributes>
          <ns2:Name>SUB ID TYPE</ns2:Name>
          <ns2:Value>MSISDN</ns2:Value>
        </ns2:Attributes>
      </ns3:AttributeList>
    </ns3:PerformFunctionQuery>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.2 Response

Example Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <prv:AttributeList xmlns:prv="http://xml.vodacom.com/schema/PRV">
      <com:Attributes xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
        <com:Name>ICAP_ID</com:Name>
        <com:Value>147673503</com:Value>
      </com:Attributes>
      <com:Attributes xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
        <com:Name>PAYMENT_METHOD</com:Name>
        <com:Value>C</com:Value>
      </com:Attributes>
      <com:Attributes xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
        <com:Name>SP_SEQ</com:Name>
        <com:Value>1</com:Value>
      </com:Attributes>
      <com:Attributes xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
        <com:Name>CALL_ROUTE_FLAGS</com:Name>
        <com:Value>00000ADEF0</com:Value>
      </com:Attributes>
      <com:Attributes xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
        <com:Name>OFFER_ID</com:Name>
        <com:Value>1BM</com:Value>
      </com:Attributes>
    </prv:AttributeList>
  </soap:Body>
</soapenv:Envelope>
```

3.3 Query Function

3.3.1 Request

Request

```
<prv:QueryFunctionQuery xmlns:prv="http://xml.vodacom.com/schema/PRV"
xmlns:com="http://xml.vodacom.com/Schema/APD/Common">
    <prv:FunctionSelector>string</prv:FunctionSelector>
    <prv:AttributeList>
        <!--1 or more repetitions:-->
        <com:Attributes>
            <com:Name>string</com:Name>
            <com:Value>string</com:Value>
        </com:Attributes>
    </prv:AttributeList>
</prv:QueryFunctionQuery>

//……………………….Example……………………….

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns3:QueryFunctionQuery xmlns:ns2="http://xml.vodacom.com/Schema/APD/Common"
xmlns:ns3="http://xml.vodacom.com/schema/PRV">
      <ns3:FunctionSelector>QPYMTH</ns3:FunctionSelector>
      <ns3:AttributeList>
        <ns2:Attributes>
          <ns2:Name>SUB ID</ns2:Name>
          <ns2:Value>0665110159</ns2:Value>
        </ns2:Attributes>
        <ns2:Attributes>
          <ns2:Name>SUB ID TYPE</ns2:Name>
          <ns2:Value>MSISDN</ns2:Value>
        </ns2:Attributes>
      </ns3:AttributeList>
    </ns3:QueryFunctionQuery>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.2 Response

Example Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <ns1:QueryFunctionResult xmlns:ns1="http://xml.vodacom.com/schema/PRV">
      <ns1:AttributeList>
        <common:Attributes xmlns:common="http://xml.vodacom.com/Schema/APD/Common">
          <common:Name>PAYMENT_METHOD</common:Name>
          <common:Value>C</common:Value>
        </common:Attributes>
      </ns1:AttributeList>
      <ns1:Party>
        <ns1:seq>0</ns1:seq>
        <ns1:icapId>0</ns1:icapId>
        <ns1:icapRole/>
        <ns1:titleId/>
        <ns1:surname/>
        <ns1:firstName/>
        <ns1:initials/>
        <ns1:langCode/>
        <ns1:birthDate/>
        <ns1:activeYN/>
        <ns1:listedYN/>
        <ns1:gender/>
      </ns1:Party>
    </ns1:QueryFunctionResult>
  </soap:Body>
</soapenv:Envelope>
```

Project Manager

Name		**Date Received:**	
Approved for Investigation?		**YES** □	**NO** □
Project Sponsor Signature			
Date Approved			
Comments:			

Project Sponsor/Owner

Name	Hazel Mokoto	**Date Received:**	
Approved for Investigation?		**YES** □	**NO** □
Authors Signature			

Date Approved	
Comments:	