INTERFACE SPECIFICATION

FOR

CONSENT BASED SSN VERIFICATION

(CBSV) WEB SERVICE

RELEASE NUMBER: 1.0

OCTOBER 2, 2008

TABLE OF CONTENTS

1.0 INTRODUCTION 3

2.0 CBSV CONTEXT FOR USE 4

3.0 PROCEDURAL PREREQUISITES 5

4.0 GENERAL TECHNICAL PREREQUISITES 6

5.0 WEB SERVICE REQUIREMENTS 7

6.0 INTERFACE OPERATIONS 8

7.0 SEQUENCE OF REQUESTS AND RESPONSES 9

8.0 STRUCTURE OF REQUESTS AND RESPONSES 13

8.1 OVERVIEW OF REQUESTS 13

8.2 PING REQUEST AND RESPONSE STRUCTURE 13

8.2.1 Request 14

8.2.1.1 PING Request SOAP Header 15

8.2.1.2 PING Request SOAP Body 16

8.2.2 Response 17

8.2.2.1 Operation Response 17

8.2.2.2 Failure Responses 17

8.3 VERIFY REQUEST AND RESPONSE STRUCTURE 18

8.3.1 Request 18

8.3.1.1 VERIFY Request SOAP Header 20

8.3.1.2 VERIFY Request SOAP Body 20

8.3.2 Response 21

8.3.2.1 Operation Response 21

8.3.2.2 Failure Responses 22

9.0 WSDL AND XML SCHEMA DEFINITION 23

9.1 CBSV SERVICES-WSDL 23

9.2 CBSVSERVICES-XSD 24

10.0 ACRONYMS 26

TABLE OF FIGURES

Figure 1 ***CBSV Web Service Ping Process Data Flow*** **10**

Figure 2 ***CBSV Web Service Verify Process Data Flow*** **11**

Figure 3 ***CBSV Web Service Verify Process Data Flow*** **12**

1.0 INTRODUCTION

The CBSV Web Service application allows verification of the name and Social Security Number (SSN) of individuals against the Social Security Administration's (SSA's) master file.

A CBSV Web Service user termed as "Requesting Party" throughout this document will need to develop a Web Service client to interface with SSA CBSV Web Service application.

This document identifies the interface specifications between a CBSV Requesting Party and the SSA CBSV Web Service. The exchange of data in the form of requests and responses between Requesting Party and SSA is described in detail. An overview of SSA's architectural components that interact with the CBSV Web Service is also provided.

2.0 CBSV CONTEXT FOR USE

The CBSV service will verify if a submitted name and SSN combination match SSA's records for any business or other entity (i.e., User) that has registered to use the service and has obtained the written consent of the number holder.

Once the Requesting Party's submission passes SSA's finance and agreement checks, CBSV returns verification results, including a 'Y' to indicate that the information submitted matches SSA's records, 'N' to indicate that the information submitted does not match SSA's records, 'D' to indicate that, according to SSA's records, the Number Holder is deceased. Specific information as to which data elements did not match SSA records will not be available.

The CBSV Web Service will provide real-time results for single name/SSN combinations, one at a time. As part of the Web Service User Login process, the Web Service client will submit the Web Service User Identifier (ID) and password, as well as the data that is to be verified. The Web Service client sends the encrypted verification request to SSA's CBSV Web Service.

3.0 PROCEDURAL PREREQUISITES

In order to be eligible to use the CBSV Web Service interface, the following CBSV registration criteria must have been met, as per SSA guidelines:

- Registration Fee ($5,000) must be paid;
- Form SSA-88 must be completed by applicant and approved by SSA;
- Form SSA-1235 (the CBSV User Agreement between SSA and the requesting party for use of CBSV) must be completed by applicant and approved by SSA; and
- Registration process must be completed via SSA's Integrated Registration Services (IRES) Business Service Online (BSO) Application to obtain a CBSV Web Service User ID and password and to activate the CBSV Web Service Role.

4.0 GENERAL TECHNICAL PREREQUISITES

The following technical prerequisites must be satisfied by the CBSV Requesting Party to use the CBSV Web Service:

- Provision of a public key of the x.509 Digital Certificate that will be used to sign the Simple Object Access Protocol (SOAP) envelope. The certificate must be from a recognized source.
- Establishment of a Web Service development environment that supports using the .NET and/or Java-based programming language technologies.

5.0 WEB SERVICE REQUIREMENTS

Development of Web Service clients for Web Service transmission requires the following technical characteristics:

- Conformance to World Wide Web Consortium (W3C) Web Service standards, including SOAP, Web Services Description Language (WSDL), and Web Service Security (WS-Security, also known as WSS).
- Presence of a transport layer security using Secure Hypertext Transfer Protocol (HTTPS), using Secure Sockets Layer (SSL) Certificates signed by well known, trusted Certification Authorities (CAs).
- Protections using the following authentication mechanisms:
  - Client authentication using the User ID /Password as part of the WSS SOAP header; and
  - Strong authentication (using x.509 Client Certificates), which authenticates the Requesting Parties based on a digital signature over the SOAP: body and a timestamp element.

6.0 INTERFACE OPERATIONS

CBSV Requesting Parties must register for a User ID and password with SSA prior to using the CBSV Web Service. They must also acquire an x.509 certificate and provide SSA with a public key for this certificate prior to using the Web Service. WSS is used in tandem with an x. 509 certificate.

Requesting Parties must register for a User ID and password via the IRES utility prior to using the web service. Requesting Parties must notify SSA of the name of the person whose Form SSA-88 will be used to request access to the CBSV Web Service.

The Requesting Party can verify one name/SSN per request; every request must be a separate web service call.

Data is exchanged between Requesting Party Application and SSA using CBSV Web Services. SSA will publish the WSDL required to secure the exchange of data.

The CBSV Web Service provides two operations:

- *Ping*: The *ping* operation is used to determine if the SSA resources are available. Requesting Party's credentials are used to authenticate the request. *(ref. Figure 1 - CBSV Web Service Ping Process Data Flow)*
- *Verify*: The *verify* operation compares the data provided by the Requesting Party with the data in the SSA master files. *(ref. Figures 2 and 3 - CBSV Web Service Verify Process Data Flow)*

7.0 SEQUENCE OF REQUESTS AND RESPONSES

The following diagrams (Figure 1 for the Ping operation, and Figures 2 and 3 for the Verify operation) show the typical sequence of requests and responses between the SSA Requesting Parties and the CBSV Web Service architectural components.

Upon receiving a request, several preliminary validations are performed. Specific responses for each validation are described in Section 8.0. Queries include the following:

- *Schema Validation* – Is the request data in accordance to the format specified in the schema?
- *Authentication* – Is the question "Who is the user and is the user really, who he/she asserts" answered successfully
- *Authorization* – Once authenticated, is the user authorized to perform the requested operation on the requested resource?

If any of the queries return a negative result, the requested operation cannot be performed.

If the queries are successful and return positive confirmations, the requested operation may be performed. Specific responses for each operation are described in Section 8.0.

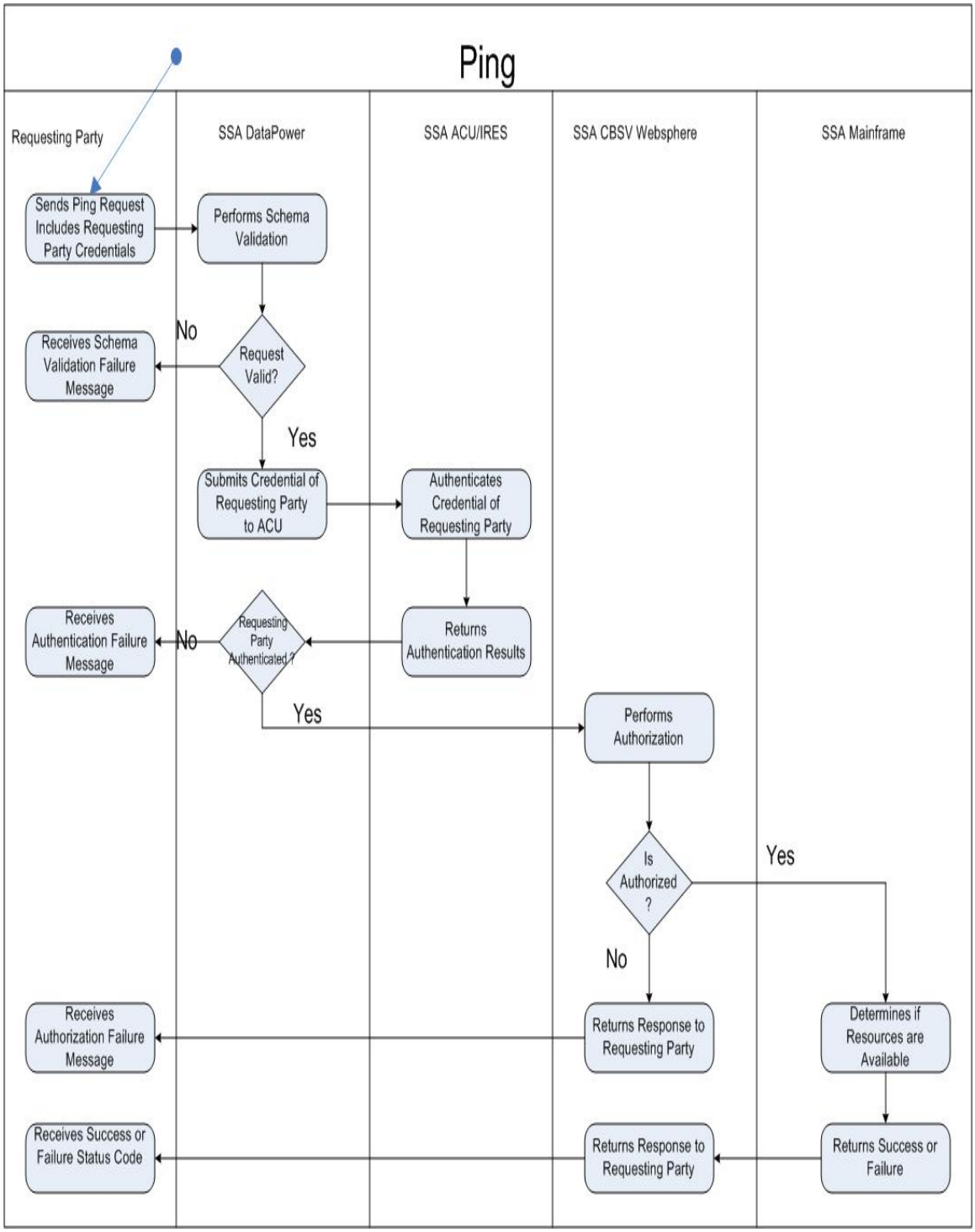

Figure 1 ***CBSV Web Service Ping Process Data Flow***

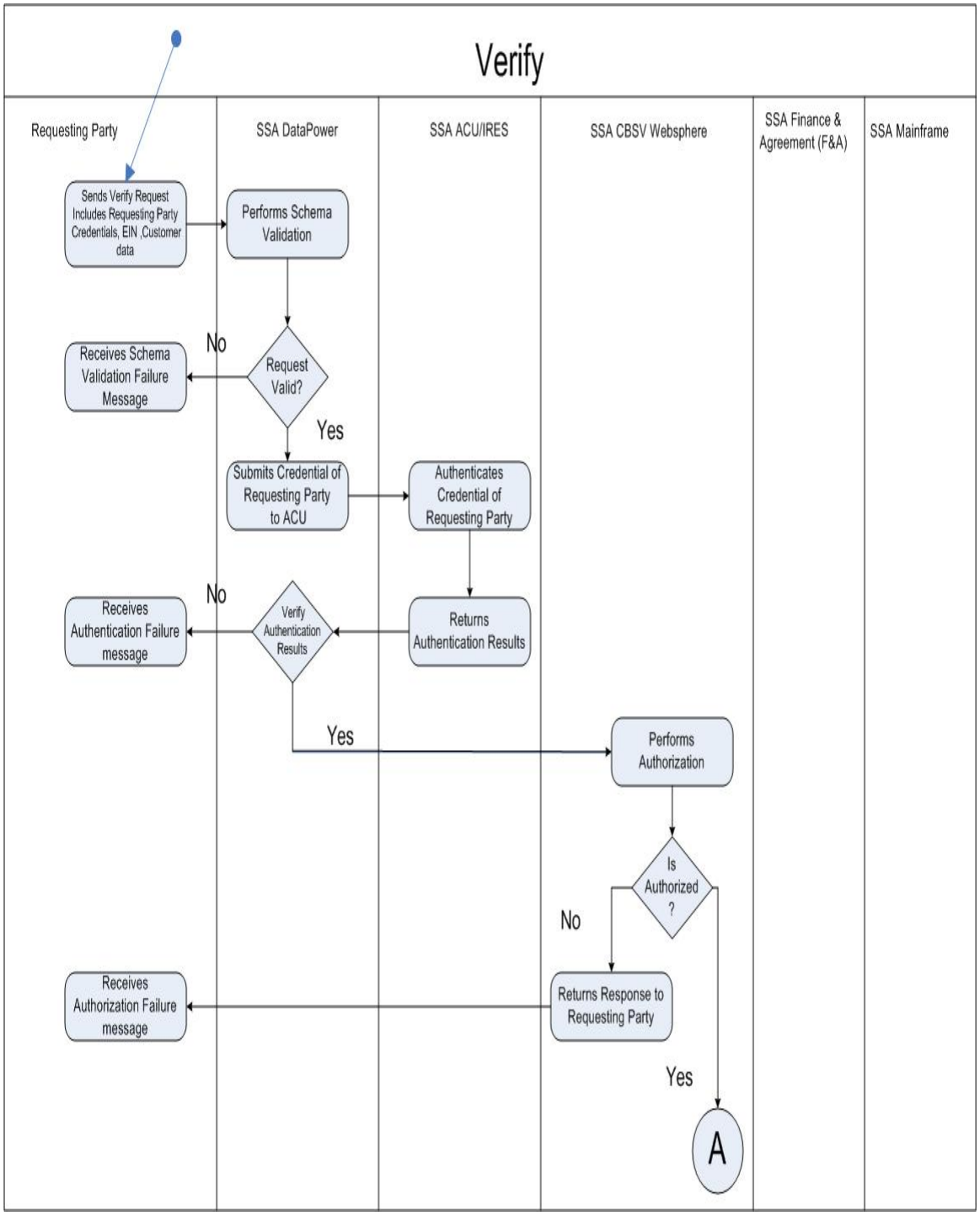

Figure 2 ***CBSV Web Service Verify Process Data Flow***

(Continued in Figure 3)

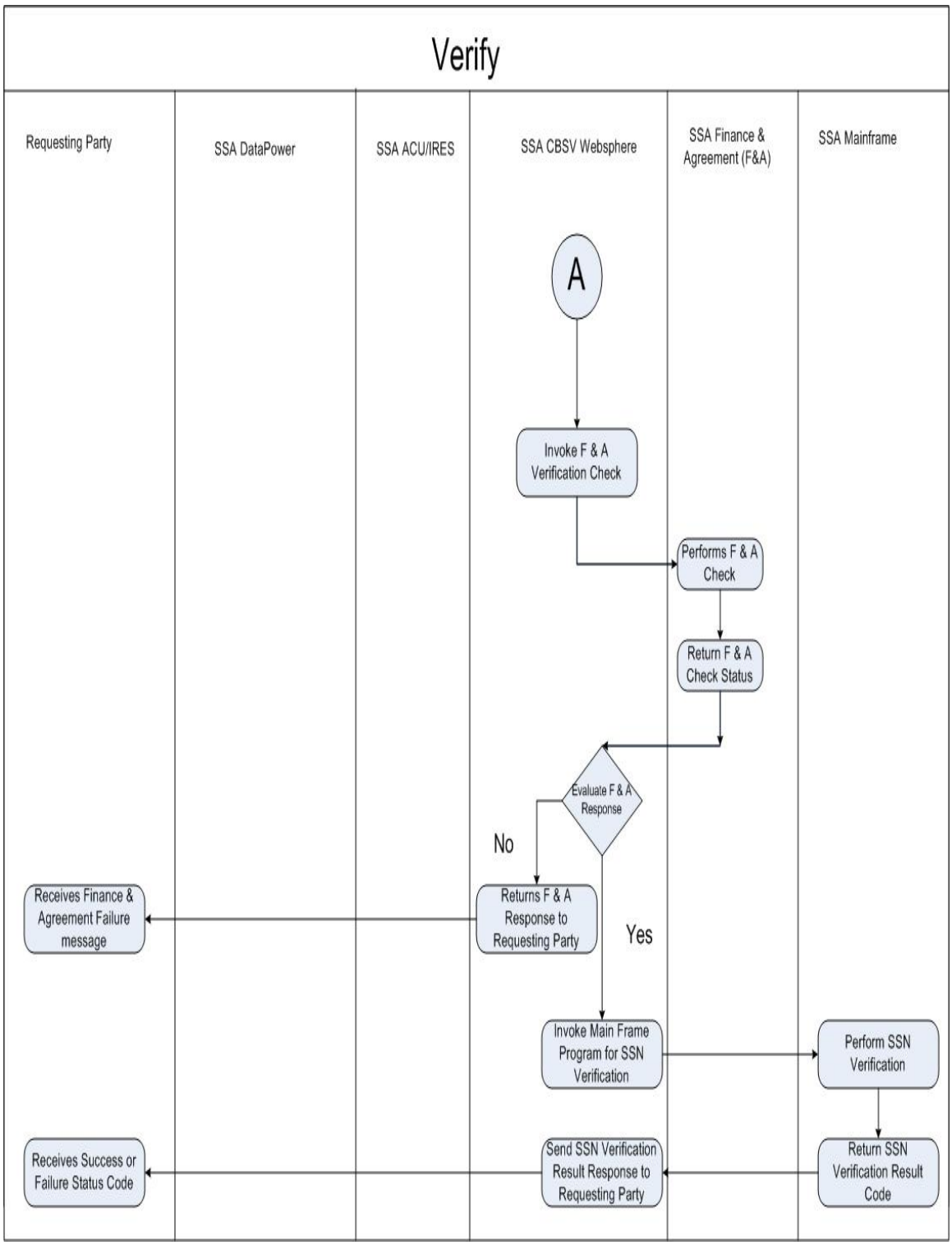

Figure 3 ***CBSV Web Service Verify Process Data Flow***

8.0 STRUCTURE OF REQUESTS AND RESPONSES

Requesting Parties must use the WSDL provided by SSA for Web Service client development. Information about how to invoke CBSV Web Services is contained in the WSDL. The WSDL is provided in Section 9.0.

SSA uses SOAP and WS-Security. SOAP headers that include credentials are required in all requests (see Overview of Requests for credentials required for a specific request in SOAP Header).

The following WS-Security options are required for securing CBSV:

- The SOAP header must conform to the "Web Services Security UsernameToken Profile." It must contain UsernameToken of type wsse: Username with an appropriate password of type wsse: PasswordText.
- The SOAP message must be signed, leveraging "Web Services Security: SOAP Message Security" guidance.
- To digitally sign the message, the Requesting Party will need an x.509 certificate from a trusted provider (for example, VeriSign, Entrust, etc.) or an internal CA. The Requesting Party will provide SSA with a public key for this certificate.

The Requesting Party will send the .cer file that contains the public key from their x.509 certificate to SSA at web.service.testing@ssa.gov. In order to allow the attachment to pass through SSA's firewall, the .cer extension should be changed to .txt, or the file can be sent using compression software with a .zip extension.

8.1 OVERVIEW OF REQUESTS

Operation	Major data elements sent with the request	Section
ping	The Requesting Party's credentials in SOAP header	8.2.1
verify	The Requesting Party's credentials in SOAP header; SSN, Name, and other optional data as *verify* parameters	8.3.1

8.2 PING REQUEST AND RESPONSE STRUCTURE

The *ping* operation is used to determine if the SSA resources are available. The Requesting Party's credentials are used to authenticate the request.

8.2.1 *Request*

The following data elements will be sent in the SOAP Header of the Ping request (see example for attributes that may be required to support these parent elements):

Element	Value Required?	Value	Example
<wsse: SecurityTokenReference >	Required	Value is the Public Key of the certificate	See example in Section 8.2.1.1
<wsse:UsernameToken>	Required	Includes 8-digit <**Username**> and <**Password**>; See example for child elements that may be required (such as Nonce, Created)	See example in Section 8.2.1.1
< ds:Signature xmlns="http://www.w3.org/2000/09/xmldsig#">	Required	See example for child elements that may be required	See example in Section 8.2.1.1

<Username> - Contained within this tag is the Requesting Party's User ID. IRES will perform validation to ensure that a User ID is submitted, and its length is 8, and that it is alpha-numeric.

<**Password**> - Contained within this tag is the Requesting Party's password. IRES will perform validation to ensure that a password is submitted, its length is 8, and that it is alpha-numeric.

8.2.1.1 PING Request SOAP Header

<soapenv: Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Header>

<wsse: Security soapenv: mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">

<ds:Signature Id="Signature-16260234" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

<ds:SignedInfo>

<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

<ds:Reference URI="#id-1934804">

<ds:Transforms>

<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

</ds:Transforms>

<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestValue>nfdqOvUMsiWz4OJva/einMCM/x0=</ds:DigestValue>

</ds:Reference>

</ds:SignedInfo>

<ds:SignatureValue>

JAJMc6Qt5K9KZssnEG8v3rts7oTr7dBjMte3XwkckMYW1wha69oFx6DqaXncjd3Uk3oKuD1BaPDD

bustMxNYXpJ7zYM/KCCWYe4j2j4PQ6rXURQ7flyAL7qzBxqHL8Eh28FxYKtwcdw780HAMR8sSSlB

WEZZd7/rIw54izU87xI=

</ds:SignatureValue>

<ds:KeyInfo Id="KeyId-18464534">

<wsse:SecurityTokenReference wsu:Id="STRId-22046976" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsse:KeyIdentifier
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"

ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier">u9LFn414C
6Fdpr1x875aHM6tVkk=

</wsse:KeyIdentifier>

</wsse:SecurityTokenReference>

</ds:KeyInfo>

</ds:Signature><wsse:UsernameToken wsu:Id="UsernameToken-16302158"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsse:Username>PECAS924

</wsse:
Username>

<wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">1newpass</wsse:Password>

<wsse:Nonce>TKIX9rZyRHr6Ao5wJGgXjw==

</wsse:Nonce><wsu:Created>2008-07-
21T14:42:25.104Z</wsu:Created></wsse:UsernameToken>

</wsse:Security>

</soapenv:Header>

<soapenv:Body wsu:Id="id-1934804"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"/>

</soapenv:Envelope>

8.2.1.2 PING Request SOAP Body

N/A

8.2.2 *Response*

8.2.2.1 Operation Response

The following status code responses will be returned:

Element	Value	Element	Value
<statusCode>	0000	<statusDescription>	Successful
<statusCode>	0151	<statusDescription>	System Failure

8.2.2.1.1 *PING* SOAP Success Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body><ns1:ActionResponse xmlns:ns1="http://cbsv.ssa.gov/ws/datatype">
    <ns1:responseCode>0000</ns1:responseCode>
<ns1:responseDescription>Successful</ns1:responseDescription></ns1:ActionResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```

8.2.2.2 Failure Responses

The following SOAP fault responses will be returned if the request does not pass schema validations, or if the user is not authenticated or not authorized:

SOAP Fault	Sample SOAP Body
Schema Validation Failure	<env:Fault> <faultcode>env:Server</faultcode> <faultstring>Schema validation failure</faultstring> </env:Fault>
Authentication Failure	<env:Fault> <faultcode>env:Server</faultcode> <faultstring>Authentication Failure</faultstring> </env:Fault>
Authorization Failure	<env:Fault> <faultcode>env:Server</faultcode> <faultstring>Authorization Failure</faultstring> </env:Fault>
System Failure	<ns1:responseCode>0151</ns1:responseCode> <ns1:responseDescription>System Failure </ns1:responseDescription>

8.3 VERIFY REQUEST AND RESPONSE STRUCTURE

The *verify* operation compares the data provided by the Requesting Party with the data in the SSA master files. The Requesting Party's credentials are used to authenticate the request. One *verify* request per customer.

8.3.1 *Request*

The following data elements will be sent in the SOAP Header of the Verify request:

Element	**Value Required?**	**Value**	**Example**
<wsse:SecurityTokenReference>	Required	Value is the Public Key of the certificate	See example in Section 8.2.1.1
<wsse:UsernameToken>	Required	Includes 8-digit <**Username**> and <**Password**>; See example for child elements that may be required (such as Nonce, Created)	See example in Section 8.2.1.1
<ds:Signature xmlns="http://www.w3.org/2000/09/xmldsig#">	Required	See example for child elements that may be required	See example in Section 8.2.1.1
<ssn>	Required	9-digit SSN	123456789
<firstName>	Required	10-character First Name	CASEY
<middleName>	Optional	7-character Middle Name	WATSON
<lastName>	Required	13-character Last Name	SONES

Element	Value Required?	Value	Example
<dateOfBirth>	Optional	MMDDYYYY	07041976
<genderCode>	Optional	F or M	M

<Username> - Contained within this tag is the user's User ID. IRES will perform validation to ensure that a User ID is submitted, its length is 8, and that it is alpha-numeric.

<**Password**> - Contained within this tag is the user's password. IRES will perform validation to ensure that a password is submitted, its length is 8, and that it is alpha-numeric.

<ssn> - Contained within this tag is a 9-digit SSN. CBSV will perform validation to ensure that a SSN is submitted, it is 9 digits long, and is within valid value ranges of SSNs issued by SSA.

<**firstName**> - Contained within this tag is a first name (no suffixes). CBSV will perform validation to ensure that a first name is included in the request, that its maximum length is 10 and that it contains only upper-case letters A to Z. Any other characters such as apostrophes, spaces, periods, and hyphens are removed and the name is compressed. If the name exceeds 10 characters, the excess characters at the end of the name are truncated.

<**middleName**> - Optionally contained within this tag is a middle name (no suffixes). CBSV will perform validation of the middle name, if it is included in the request, to verify that, its maximum length is 7 and it contains only upper-case letters A to Z. Any other characters such as apostrophes, spaces, periods, and hyphens are removed and the name is compressed. If the name exceeds 7 characters, the excess characters at the end of the name are truncated.

<**lastName**> - Contained within this tag is a last name (no suffixes). CBSV will perform validation to ensure that a last name is included in the request, that its maximum length is 13, and that it contains only upper-case letters A to Z. Any other characters such as apostrophes, extra spaces, periods, and hyphens are removed and the name compressed. If the name exceeds 13 characters, the excess characters at the end of the name are truncated.

<dateOfBirth> - Optionally contained within this tag is the 8-digit string representing the month of birth in the format MM, day of birth in the format DD, and year of birth in the format YYYY (i.e. MMDDYYYY). IRES will perform validation of the date, if included in the request, to verify the format and value ranges for the date components. It will to verify that the month (MM) is in the range of 01 to 12. It will verify that a day is submitted if a birth date is included in the request, and the day (DD) is in the range of 01 to 31. And it will verify that the year (YYYY) is in the range of 1900 to "current year minus 12". There shall be error handling/validation to avoid incongruous date information, e.g., April 31, Feb 29 on non-leap years, etc.

<**genderCode**> **-** Optionally contained within this tag is the 1-digit character F for Female or M for Male.

8.3.1.1 VERIFY Request SOAP Header

See Section 8.2.1.1.

8.3.1.2 VERIFY Request SOAP Body

```
<soapenv:Body wsu:Id="id-7644051"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<dat:VerifyRequest>
  <dat:ssn>011868502</dat:ssn>
  <dat:firstName>CASEY</dat:firstName>
  <dat:middleName/>
  <dat:lastName>SONES</dat:lastName>
  <dat:dateOfBirth/>
  <dat:genderCode>F</dat:genderCode>
</dat:VerifyRequest>
```

8.3.2 *Response*

8.3.2.1 Operation Response

The following status codes will be returned:

Element	Value	Element	Value
<statusCode>	0000	**<statusDescription>**	Successful
<statusCode>	0001	**<statusDescription>**	Verification successful, but this person is deceased
<statusCode>	9991	**<statusDescription>**	Verification unsuccessful
<statusCode>	9910	**<statusDescription>**	Agreement in force: Negative account balance
<statusCode>	9920	**<statusDescription>**	Agreement in force: No account found
<statusCode>	9930	**<statusDescription>**	Agreement in force : Unable to check account balance
<statusCode>	9940	**<statusDescription>**	Agreement not in force
<statusCode>	9950	**<statusDescription>**	Agreement not in force : Negative account balance
<statusCode>	9960	**<statusDescription>**	Agreement not in force: No account found
<statusCode>	9970	**<statusDescription>**	Agreement not in force: Unable to check account balance
<statusCode>	9980	**<statusDescription>**	No agreement found: Unable to check account balance
<statusCode>	9990	**<statusDescription>**	Systems problem: API not functioning or network unavailable
<statusCode>	0151	**<statusDescription>**	System Failure

8.3.2.1.1 Verify Success SOAP Body

```
<ns1:responseCode>0000</ns1:responseCode>
<ns1:responseDescription>Verification Successful
</ns1:responseDescription>
```

8.3.2.2 Failure Responses

The following SOAP fault responses will be returned if the request does not pass schema validations, or if the user is not authenticated or not authorized:

SOAP Fault	**Sample SOAP Body**
Schema Validation Failure	<env:Fault> <faultcode>env:Server</faultcode> <faultstring>Schema validation failure</faultstring> </env:Fault>
Authentication Failure	<env:Fault> <faultcode>env:Server</faultcode> <faultstring>Authentication Failure</faultstring> </env:Fault>
Authorization Failure	<env:Fault> <faultcode>env:Server</faultcode> <faultstring>Authorization Failure</faultstring> </env:Fault>
Verification Failure	<ns1:responseCode>9991</ns1:responseCode> <ns1:responseDescription>Verification unsuccessful</ns1:responseDescription> </ns1:ActionResponse>

9.0 WSDL AND XML SCHEMA DEFINITION

9.1 CBSV SERVICES-WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://www.ssa.gov/cbsv/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://www.ssa.gov/cbsv/"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:impl="http://cbsv.ssa.gov/ws/datatype"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xsd:schema>
  <xsd:import namespace="http://cbsv.ssa.gov/ws/datatype"
schemaLocation="CBSVServices.xsd"/>
  </xsd:schema>
  </wsdl:types>
  <wsdl:message name="verifyRequest">
    <wsdl:part name="verifyRequest" element="impl:VerifyRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="verifyResponse">
    <wsdl:part name="verifyResponse" element="impl:ActionResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="pingResponse">
    <wsdl:part name="pingResponse" element="impl:ActionResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="pingRequest">
  </wsdl:message>
  <wsdl:portType name="CBSVServicesPortType">
    <wsdl:operation name="ping">
      <wsdl:input message="tns:pingRequest" wsaw:Action="wsdl:ping">
    </wsdl:input>
      <wsdl:output message="tns:pingResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="verify">
      <wsdl:input message="tns:verifyRequest"
wsaw:Action="wsdl:verify">
    </wsdl:input>
      <wsdl:output message="tns:verifyResponse">
    </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="CBSVServicesSoapBinding"
type="tns:CBSVServicesPortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="ping">
```

```
      <soap:operation soapAction="ping" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="verify">
      <soap:operation soapAction="verify" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="CBSVServices">
    <wsdl:port name="CBSVServicesPort"
binding="tns:CBSVServicesSoapBinding">
      <soap:address
location="http://localhost:9080/CBSVWebService/services/CBSVServices"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

9.2 CBSVSERVICES-XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://cbsv.ssa.gov/ws/datatype"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://cbsv.ssa.gov/ws/datatype">
    <xsd:element name="ActionResponse">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="responseCode" type="xsd:string"/>
                <xsd:element name="responseDescription"
type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="VerifyRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ssn">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:pattern value="[0-9][0-9][0-9][0-9][0-
9][0-9][0-9][0-9][0-9]"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="firstName">
                    <xsd:simpleType>
```

```
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="10"/>
                            <xsd:minLength value="1"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="middleName" nillable="true">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="7"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="lastName">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="13"/>
                            <xsd:minLength value="1"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="dateOfBirth" nillable="true">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:pattern value="\d{8}|\d{0}"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="genderCode" nillable="true">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:pattern value="[mfMF]?"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

From October 27, 2008, onward, for the most recent Consent Based SSN Verification Web Service WSDL file refer to the following link:

https://ws.ssa.gov/CBSVWS/services/CBSVServices?wsdl

Until that time, the Consent Based SSN Verification Web Service WSDL and XSD files can be downloaded from links found at www.socialsecurity.gov/cbsv

10.0 ACRONYMS

The following list defines the acronyms used throughout this document.

Acronym	Acronym Definition
API	Application Programming Interface
BSO	Business Services Online
CA	Certification Authorities
CBSV	Consent Based SSN Verification Project
HTTPS	Secure Hypertext Transfer Protocol
IP	Internet Protocol
IRES	Integrated Registration Services
SOAP	Simple Object Access Protocol
SSA	Social Security Administration
SSL	Secure Sockets Layer
SSN	Social Security Number
User ID	User Identifier
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
WSS	Web Service Security
XML	eXtensible Markup Language
XSD	XML Schema Definition