Appendix Q: Web Services

This appendix documents how the web service technologies SOAP and WSDL are leveraged by the SIF standard to define a second reference transport for conveying SIF messages between agent/applications and the ZIS.

While it is dependent upon and references details contained in the Architecture, Messaging, Infrastructure, and Zone Services sections, unless otherwise noted, those sections remain independent of the information contained here. It is anticipated that the contents of this Appendix will be more fully integrated into those sections in subsequent releases.

Q.1 Introduction and Background

The overarching goal of the web servicemappingof the SIF Transport was to insure that the large and growing number of deployed SIF-based solutions (Zones), which were created independently of these technologies, could stillincorporate them effectively in a seamless and incremental fashion without impacting day to day operations.

This subsection provides the context for understanding how that goal was achieved.

Q.1.1 Guiding Principles

The set of guiding principles below determined how the SIF architecture was extended to support web service technology. This included the addition of a new SOAP-based SIF reference transport and a set of WSDL port types to encapsulate the services provided by the ZIS.

Q.1.1.1 Backwards compatibilitymust not be broken

There is a seamless and incremental web service migration path provided for all existing deployed SIF v2.x Zones.

- A “Web enabled” ZIS (v2.5 and above) is capable of supporting all v2.x agent/application pairs.
- SIF Web Applications using the new technologies are capable of being added to a SIF v2.5 and above Zone without impacting the operations of any pre-existing component.
- Existing SIF Zone components remain completely unaware of whether their “partner” (requestor, responder, publisher or subscriber) is a SIF Web Application or not.
- SIF Web Applications conforming to the architectural requirements defined in this specification are capable of transparently replacing any equivalent agent/application pair in the Zone ...and vice-versa.

Q.1.1.2Extend the architecture by embracing the new technologies, not by replacingor patching existing ones.

Web service technologies such as SOAP and WSDL provide a second reference infrastructure for the SIF Standard.

- They co-exist with and do not replace the existing SIF infrastructure within a SIF Zone.

- Their usage is “logically consistent”, and not simply a “wrapper” placed around the existing SIF infrastructure.

Components conforming to either infrastructure continue to be equal “citizens” in all SIF v2.5 and above Zones.

Q.1.1.3Incorporating a second reference transport within the SIF standard must not decrease “Out of the Box” application interoperability.

Theset of requirements placed on the new transport is as proscriptive as those on the original. Wherethe functionality of an architectural style and a web service standard overlap (ex: REST and SOAP), or two versions of the same web service standard are incompatible to any degree (ex: WSDL 1.1 and WSDL 2.0) only one choice wasmade a normative dependency for the SIF standard

In those cases wheremultiple incompatible options exist within the same version of a web service standard (ex: SOAP “literal” vs. SOAP “encoded”) only one option was made a normative dependency for the SIF standard.

Q.1.2 Glossary of Terms

The following terms will be used throughout the rest of this appendix. Wherever possible, they reflect common industry usage and consensus.

Q.1.2.1 Service

A Service is a software application that responds to requests made of it by client applications. Any given application can be both a service used by multiple clients, and a client which itself uses other services.

Every service possesses a public interface, defining exactly what operations its clients can ask it to do. This interface specifies the methods the service supports, the data these methods accept and the results they return. Each service also has a hidden (private) implementation which determines how it will actually “service” these requests.

The fact that the service implementation is hidden (encapsulated) means that even if the details of that implementation radically change, if the interface is unaffected, none of the clients of that service will be impacted. Having clients be independent of how a service is actually implemented is a key enabler of the architecture described in this appendix.

Q.1.2.2 SIF Object Service

A SIF Object Provider can be considered as an “Object Service”. The Service interface in this case is composed completely of CRUD (Create, Read, Update, and Delete) operations for the object data it provides. Making a SIF Request is equivalent to invokinga “Read” operation, and the SIF Event equates to the “Create”, “Update” and “Delete” service operations.

Neither the cross-object relationships within the SIF object hierarchy nor the behavioral aspects of an educational process are encapsulated by these Object Services.

Q.1.2.3 SIF Zone Service

Starting in SIF v2.4, the existing SIF infrastructure was extended to support “non-CRUD” operations, which allowed Zone Services to be constructed which encapsulate both the details of the object hierarchy and associated transactional behavior (see Section 7).

Three new SIF message types (ServiceIn, ServiceOut and ServiceNotify) were required to carry the "non-object" operation requests, responses and event notifications respectively. As a result, any client of a Zone Service MUST support this extended SIF infrastructure.

All of the message types of the original SIF infrastructure (including these new Zone Service extensions) have been mapped to their SOAP equivalents.

Q.1.2.4Web Service

A Web Service is a Service that conforms to the following general requirements.

- The format of the data it accepts and produces is defined by XML Schema.
- Its defined operations (interface) are described by the Web Services Description Language (WSDL) and automatically generate "invocation stubs" in clients of that service.
- Each operation is "bound" to a specific XML schema which defines the contents of the associated data.
- Web clients interact with the web service in a manner prescribed by its WSDL description. They exchange data in formats defined by its bound XML schemas, carried over the SOAP transportin accordance with a set of conventions defined in the WS-I Basic Profiles (BPs) and Basic Security Profiles (BSPs).
- A family of additional WS-* standards provide the conformant Web Service with many of the messaging capabilities already provided by the Zone Integration Server (ZIS) for SIF applications within the Zone. These capabilities include reliable message delivery, content based routing, and automatic service discovery.

Q.1.2.5 SIF Infrastructure Web Service (SIWS)

The SIWSisa web service(represented by a WSDL file) with a fully defined set of interfaces (WSDL Port Types) each consisting of a set of operations and an implied choreographyfor operationinvocation. These interfacesMUST be provided and supported by all "web enabled" SIF v2.5 and above Zone Integration Servers.

The SIF Infrastructure Web Serviceprovidesits web clients with access to the complete range of existingZIS functionality. Any client of this web serviceMUST be capable of being a full participant in the SIF Zone, without maintaining any dependency on the actual ZIS implementation behind these interfaces.

Q.1.2.6 SIF Web Application (SWA)

A SWA is the web client of the SIF Infrastructure Web Service, and it MUST be able to invokeSIWS operations over the SOAP transport in a manner completely analogousto how a SIF Agent/Application invokes ZIS methods over the HTTP/S SIF transport.

A SWA accesses the SIF Infrastructure Web Serviceover SOAP in accordance with the SIWSdefined set of WSDL interfaces, and it can be developed using standard web service toolkits.

If a SWA replaces any existing SIF v2.x Agent/Application pair,such a substitution will be (at the infrastructure level) invisible to any other components of the Zone, excluding the ZIS.This is true even if it replaces an Object Provider.

A "Pull Mode" SWA will be a pure web client, and will synchronously request each new message from the SIWS. A "Push Mode" SWA will be a pure web service, implementing the supplied SIF-standard WSDL, which defines separate methods for asynchronously receiving incoming Event, Request, Notify and

ServiceInmessages, relayed by the ZIS. The Push Mode SWA supplies its (call back) service end point to the SIWS in the SIF_Registeroperation.

Q.1.3 Architectural Components

As indicated above, aSIF Web Application written to utilize the SOAP transport, and utilizingthe set of SIF Infrastructure Web Service interfaces and implicit operation invocation choreography, MUST be able to

- Participate fully in the SIF Zone
- Interoperate seamlessly on an infrastructure level with the ZIS, other SWAs, and all agent/application pairs which utilize the original HTTP/S infrastructure.

This is illustrated in the following diagram, which will be explained in further detail in the subsections below.

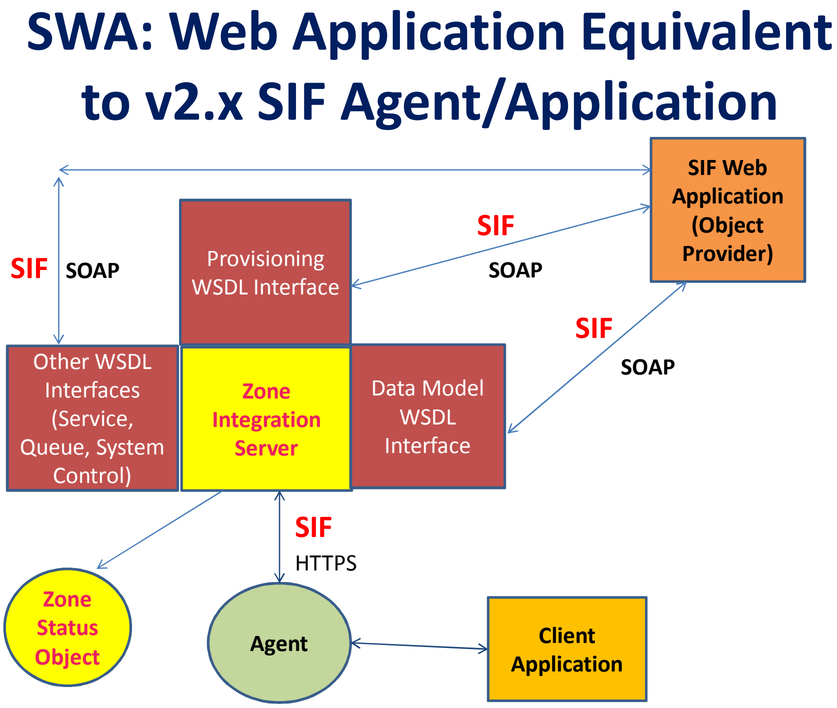
SWA: Web Application Equivalent to v2.x SIF Agent/Application
SIF Web Application (Object Provider)
SIF
SOAP
Provisioning WSDL Interface
SIF
SOAP
SIF
SOAP
Other WSDL Interfaces (Service, Queue, System Control)
Zone Integration Server
Data Model WSDL Interface
SIF
HTTPS
Zone Status Object
Agent
Client Application

Figure 1: Example of the SIF Infrastructure Web Service(SIWS) transparently and simultaneously supportingboth a“Pull Mode” SIF Web Application (SWA)over SOAP, and an “original” SIF Agent/Application component over the original HTTPS transport.

Q.1.3.1 SWA Required Capabilities

The functions provided by the current SIF Agent component are reflected in the SIF Web Application. The SWA MUST contain logic which:

- Invokes the SIF Infrastructure Web Service WSDL operations over the SOAP transport.
- Uses/expects the SIF XML message schema appropriate to the operation invoked
- Operates in conformance with the existing Agent/ZIS choreography.

The basic set of capabilities provided to a SIF Web Application by the SIWS MUST allow it to:

- Register and Provision itself in the Zone
- Get information about the other applications previously registered in the Zone
- Request SIF object data and receive valid (and understandable) Responses from the Object Provider for the object type selected.
- Subscribe to and receive Events for one or more SIF object types.
- Serve as the Object Provider for one or more object types. This includes receiving all posted Requests for object data and havingall providedResponses routed back to the correct issuing client.
- Publish Events (whether or not the SWA is serving as the Object Provider for that object type).
- Function as a Zone Service, receiving ServiceIn and issuing ServiceOut and ServiceNotify messages
- Function as a client of a Zone Service, issuing ServiceIn and receiving ServiceOut and ServiceNotify messages.
- Support the existing SIF message packetization functionality for those message types where it is required.
- Support Directed Request, Event, ServiceInand ServiceNotify messages to a specified recipient.
- Function even though it cannot receive incoming HTTP/S connections (i.e. is operating in Pull Mode).

Q.1.3.2 ZIS Optional Capabilities.

Support for the following SWA capabilities MAY be provided, but is not required of the ZIS.

Interoperability with Agent/Applications from earlier versions: Backward compatibility can optionally be extended to include interoperability (on an infrastructure level) with SIF v1.5 components and earlier.

Support for SMB: SMB support for SIF Web Applications is not required in this release. The ZIS MAY optionally provide support within the SOAP transportdefined in this appendixfor the “Intermediate” ACK, which allows an agent to block Events while awaiting a Response to an earlier Request.

Q.1.3.3 Migration Requirements

Any v2.x deployed SIF Zone can be made “web enabled” by upgrading the ZIS to a version (v2.5 or later) that supports the SOAP/WSDL mapping described in this appendix.

No other agent/application component changes need be made. Everything interoperates exactly as before. The difference is that SIF Web Applications can now register, and freely interoperate within the Zone, because the SIF v2.5 ZIS MUST support both reference transports.

Q.2 Web Service Framework

The following set of web standards, versions and options are used by SIF Web Applications to exchange XML documents with the ZIS. The collection of these normative dependencies is referred to as the Web Service Framework. All ZIS and SWAs MUST support this framework.

The reasoning behind theirindividual selection is described in an accompanying document.

Component	Choice	Options
Transport	SOAP 1.1	Document/Literal
Interface Language	WSDL 1.1	
Basic Profile	BP 1.2	WS-Addressing
Security	Transport Layer Security (TLS) Underlying protocol: HTTPS 1.1	Mutual Authentication mode supported by X.509 Certificates

Q.3 SIF HTTP/S Infrastructure to SOAP Mapping

Applications supporting two dissimilar transports can be made to interoperate by inserting a “transport bridge” between them, which bi-directionally intercepts each message and forwards it along over the new transport to the intended recipient.

The ZIS is the intermediary target of every message sent between SIF applications, and MUST bi-directionally map between the existing SIF HTTP/S infrastructure elements and the SOAP header/body message parts. This enables web applications utilizing SOAP to transparently exchange SIF object data with previously certified SIF applications - with neither side being aware of the intermediate ZIS-provided Bridge.

All SIF messages sent over the SOAP transport are divided into a SOAP Header and SOAP Body. The SOAP Headerprovides a physical location in the message structure for communicating Quality of Service (QoS) specifics such as reliability, addressability, and security as well as custom specifications. The SIF message has:

- A normative dependency on the WS-Addressing standard. It contains the required set of wsa: elements (wsa:To, wsa:From, etc.).

- A relocated SIF_Header element which provides additional “routing type” information that was formerly located within aSIF message schema (such as SIF_Event@ObjectName).

The SOAP Body contains a single SIF message, of a type and form which is completely specified by the SOAP Header and the corresponding SIF message schema for the SOAP transport.

The following sections detail the complete SOAP Transport mapping.

Q.3.1 Newset of SIFMessage Schema equivalents for the SOAP Transport

A completely parallel hierarchyto theHTTP/HTTPS SIF_Messagesubschemas (ex: SIF_Event, SIF_Register) is defined for messages being transported over SOAP.Each subschema appears in modified form in its SOAP equivalent. This was done for the following reasons:

- Information in the SOAP Header describes what type of message is being conveyed, and how to route it. The SOAP Body contains the specific operation input or output data. The SOAP Envelope therefore replaces the single top level SIF_Message container, which has been eliminated.

- Elements related to the routing or interpretation of a SIF message (including but not limited to those contained in complex elements like SIF_Response/SIF_Header and any associated packet control) have been moved out of the message schemas into a separate SIF complex element (SIF_Header) located within the SOAP Header.

  While thisheader information will not always prevent the ZIS from needing to examine the SOAP Body before routing the SOAP message (for example when supporting XML Filtering requirements, or where payload version conversion needs to be performed), itSHOULDeliminate that necessity in the remaining cases. It also providesa much more “SOAP-natural” mapping for thedata being exchanged.

Q.3.1.1 Separation of Data Model and SOAP Messaging infrastructure

The ZIS and PushAgent WSDL files, SOAP binding and accompanying SOAP Transport message schemas are now decoupled from (independent of) the SIF Data Model. This means they can be used without change to support multiple locales (US, UK, AU), State-specific Data Profiles or new versions of the Data Model itself (new objects and / or elements).

Q.3.1.1.1 High Level Changes

This required the following high level changes to the previous hierarchy of Message Schemas.

High Level Schema Change	Details
Infrastructure Data Objects	ZoneStatus and AgentACL are the two objects which belong to infrastructure, and are specifically requested and returned through separate message types (rather than the Request / Response sequences). Their SIF_Metadata element is pruned of Data Model-specific subelements such as EducationFilter.
Data Object "Schema Choices"	The elements which provide a "choice" spanning multiple members of the set of Data Object schemas (SIF_ObjectData, SIF_EventObject, ReportDataObjectand SIF_ResponseObjects) are redefined to be of SIF_BodyType. SIF_BodyType is an "any", originally created to decouple the Zone Service operation schemas from the specific Zone Service data schemas, and SIF_LogEntry from the schema of the object being logged. It plays the same role for Web Service operations. Information *about* the Event Object, Report Data Object and Response Object is all contained in the SOAP Header, while the appropriate "xs:any" element above is carried in the SOAP Body. This change means adding or deleting elements or object types from the SIF Data Model will not impact existing SOAP message schemas. Only the content of the SOAP Body on the wire will be changed.
Data Object "Name Enumerations"	The elements which provide an enumerated list of SIF object names (SIF_Provide/ObjectNames, SIF_Request/ObjectNames, SIF_Subscribe/ObjectNames, SIF_LogObjectName) are redefined to ObjectNameType (an unqualified XML name type currently used in SIF_AgentACL and SIF_ZoneStatus). The SIF_Serviceand SIF_Operation enumerated list of Object names are redefined to ServiceNameType and OperationNameType respectively, which are also unqualified XML name types. This change means adding or deleting object types will not impact any enumerated list of such types within the SOAP message schemas.
SIF_Header	The SIF_Header element found in all SIF HTTP/HTTPS messages has been expanded and moved to the SOAP Header. It encapsulates all SIF-related elements needed to understand how to process and route a SIF message. In most cases the information it contains will NOT be replicated in the Message schemas defining what is carried in the SOAP Body.

“SIF_” Prefix	All SIF_Header subelements have dropped their “SIF_” prefix (so for example the “SIF_Timestamp” element is conveyed as “Timestamp”). Any SIF message element which is the direct child of the SOAP Body has dropped its “SIF_” prefix. For example, a SIF_Event element is posted by issuing a WSDL “Event” operation with the “Event” element as the top child of the SOAP Body. All subelements within the “Event” element retain the “SIF_” prefix.
Zone Service Messages	The SOAP Body directly carries the SIF_Body subelement (an xs:Any) of the SIF_ServiceInput, SIF_ServiceOutput or SIF_ServiceNotify operations, under the corresponding name “ServiceInput”, “ServiceOutput” or “ServiceNotify”. All other elements of HTTP/S SIF_Service operations (SIF_ServiceMsgId and SIF_Operation, along with elements SIF_MaxBufferSize, SIF_Error and SIF_Version) appear in the SIF_Header.
Other	“HTTP-SOAP1.1 and “HTTPS-SOAP1.1” are added as enumerated values inDefinedProtocolType. A new “WSDLMessageNameType” contains the enumerated list of message names for the messages exchanged over the SOAP Protocol.

Q.3.1.1.2 Message Type-specific Changes

The following illustration contrasts the top level element mapping between a SIF Response message being sent across each of the supported transports.

HTTPS Transport	**SOAP Transport**
SIF_Message SIF_Response SIF_Header Header Sublements Packet control elements SIF_ObjectData (xs:choice) (Chosen Data Object Schema)	SOAP Envelope SOAP Header WS-Addressing elements SIF_Header Header Subelements Packet control elements SOAP Body QueryResponse (xs:any) (Any Data Object Schema)

The SOAP Transport example reflects the following level of decoupling:

- The SIF_Header contains those elements necessary to successfully process and route the QueryResponsemessage being conveyed
- The SOAP Body contains the SIF message QueryResponse schema as its top child element
- The SIF QueryResponse schema is independent of the actual SIF data object or SIF data elements being conveyed.

Q.3.1.2 Namespaces

The XML Namespace for the entire set of data model-independent message schemas carried within the SOAP Body in this release is: http://www.sifassociation.org/messages/soap/2.x

The XML Namespace for the SIF elements contained within the SOAP Header for this release is: http://www.sifassociation.org/transport/soap/2.x

Q.3.2SIF_Header elements within the SIF Message SOAP Header

The SOAP Header for a SIF message contains the set of elements mandated by the WS-Addressing standard, and the complex element “SIF_Header”. The SIF_Header contains subelements which provide information either required by the ZIS to route a SIF message, or by a SIF application to understand the message that has arrived.

These subelements are either general (common to all SIF messages on the SOAP transport) or they contain information specific to one or more message types (ex: “Query”) and are conditionally present only in the SIF Header of a message of that type.

There is also an “ExtendedHeaderContent” element within SIF_Header which is of xs:anyType, and may be used to carry any SIF-specific extensions (such as SAML assertions). This subelement is otherwise not defined further in this specification.

Q.3.2.1General Messaging Information

Each of these elements can be optionally contained in the SOAP Header part of every SOAP message being exchanged between a ZIS and an SWA, and some are required to be present in every such message. Where equivalent subelements in this section exist within the original SIF_Message schema hierarchy they are removed in their SOAP Transport equivalents.

Note: Any elements identified after the phrase “equivalent to” in the tables below refer to the original HTTPS SIF Message schemas.

New SIF_Header subelements contained in the SOAP Header	**Char**	**Usage/Meaning/Equivalent in HTTPS Transport Message Schema**
TransportVersion	M	Version of theSIF message to SOAPmapping this SOAP Header is conformant with. It is a string value restricted to the format “#.#.#”. Set to “2.5” for this release The rightmost digit is optional and represents the revision number (so a 2.5r1 release of the SOAP Transport would be represented as 2.5.1). .

DataModel	M	Data Model identifier which defines the schema for the set of XML elements contained in the SOAP Body payload of this message. In this release it is a URI of the form: http://www.sifassociation.org/{locale}/datamodel/2.x Locale value is one of the following; “us”, “uk”, “au”
DataModelVersion	O	Version of the actual Data Model being carried in the SOAP Body. It is equivalent to the SIF_Version attribute contained in SIF_Message. It is a string value restricted to the format “#.#.#”. Set to “2.5” for this release The rightmost digit is optional and represents the revision number (so a 2.5r1 release of the DataModel would be represented as 2.5.1). .
ZoneId	C	URI which uniquely identifies the Zone containing both the Sender and Receiver of this SOAP message. By convention it will be “.” delimited, and has one of two possible forms: • uuid:aaaa, where “aaaa” is the hexadecimal value of a globally unique Zone Identifier. Essentially this leaves the ZoneID value “opaque” • sif:xxx.yyy.zzz where xxx.yyy.zzz is a structure that reading left to right starts with most specific identification such as school and works rightward to identify the higher levels. (Ex*:sif:AcmeMiddleSchool1.CoyoteDistrict.Arizona*) In the 2nd case, the initial (most specific) field SHOULD be identical to the SIF_ZoneId attribute returned in SIF_ZoneStatus (Ex: *AcmeMiddleSchool1*) This element will be mandatory for all messages except the initial Registration by the Agent (Push or Pull). It will be returned by the ZIS in the Status message following a successful Registration, and will be included in Zone Status data.
SIF_Security	O	Equivalent to the SIF_Security element in the HTTP/S Transport SIF_Header A complex element which allows an originating agent to specify security requirements that the ZIS must ensure upon delivery of the message to recipient agents.
Timestamp	M	Equivalent to the SIF_Timestamp element in the HTTP/S Transport SIF_Header Time of message creation.

MsgId	M	Equivalent to the SIF_MsgId element in the HTTP/S Transport SIF_Header. The value is unique for each SIF message in the Zone. While the wsa:MessageId element is optional in the WS-Addressing standard, when present in the SOAP Header, the value of that element MUST equal MsgId.
MsgName	M	One of the set of enumerated names defined in WSDLMessageNameType , which corresponds to the SOAP Body of the current message.
SourceId	M	Equivalent to the SIF_SourceIdelement in the HTTP/S Transport SIF_Header. For all ZIS issued messages this value will be the ZoneId attribute returned in the SIF_ZoneStatus message, which is placed in the SourceId of every message sent by the ZIS. For SWAs, it is equivalent to the value placed in the SourceId of every message they send, which MUST match up with the internal tables of the ZIS.
DestinationId	O	Equivalent to and follows the rules for SIF_DestinationIdin the HTTP/S Transport SIF_Header. It is used by the ZIS to content-route the message based upon matching it with a pre-stored SIF_SourceId. For SWAs it is set only if the message is a "Directed Event" or "Directed Request" where the client instead of the ZIS determines the actual recipient.
SIF_Contexts	O	Equivalent to the SIF_Contexts element in the HTTP/S Transport SIF_Header. The list of Contexts to which the message applies. Currently only the default value is defined. This element was included only for compatibility with the HTTP/HTTPS transport. Actual support for any contexts conveyed in this element is optional by both Agent **and** ZIS.
PacketData	O	Complex element present whenever the SOAP message is a packet in a larger SIF Message. It allows the SOAP transport to assign a unique Message ID to each packet, while providing enough additional information to allow complete reconstruction of the Message Packet when bridging back to the HTTPS transport.
PacketData/InitialMsgId	M	The SOAP MsgId for the first packet of this message. If this is the first packet, this value MUST duplicate wsa:MessageId
PacketData/PacketNumber	M	xs:positive integer corresponding to the packet number
PacketData/MorePackets	M	xs:token with value of YES or NO

The following XML instance fragment shows the part of the SIF_Header within the SOAP Header, for aEvent Operation being invoked on the Student Admin SIF Web Application in the Acme Middle School Zone.

```
<SIF_Headerxmlns="http://www.sifassociation.org/transport/soap/2.x"soap:mustUnderstand="1">
<TransportVersion>2.5</TransportVersion>
<DataModel>http://www.sifassociation.org/us/datamodel/2.x</DataModel>
<DataModelVersion>2.5</DataModelVersion>
<ZoneId> AcmeMiddleSchool1.CoyoteDistrict.Arizona.US</ZoneId>
<SIF_Security>
<SIF_SecureChannel>
        <SIF_AuthenticationLevel>3</SIF_AuthenticationLevel>
<SIF_EncryptionLevel>4</SIF_EncryptionLevel>
</SIF_SecureChannel>
</SIF_Security>
<Timestamp>2010-10-24T15:58:33.984Z</Timestamp>
<SourceId>AcmeZIS</SourceId>
<DestinationId>StudentAdmin</DestinationId>
```

Q.3.2.2Message Type-Specific Information

The following table describes the changes to HTTP/HTTPS message schemas on a per-message type basis.

SIF_Message Component with any WSDL operation equivalents in "(…)"	SIF SOAP transport mapping
SIF_Message	SIF_Messagehas been removed. Lower level message elements form the payload of every SOAP Body. "Namespace" value has been replaced by one reflecting this SOAP mapping. "Version" is carried in the SIF_Header element now in the SOAP Header
SIF_Ack	SIF_ACK spans the cases where either a SIF_Status (success) or SIF_Error (Error) is being returned. "Error" and "Status" are top level WSDL operations (see below) SIF_ACK has been removed. "OriginalSourceId" and "OriginalMessageId are both conditionally in the new SIF_Header when either of those operations is invoked.
SIF_Error (SOAP Fault or UserLevelError)	SIF_Errorspans the cases where an immediate (synchronous)transport error for an operation just issued is being reported, or an asynchronous user level error is being reported in response to an earlierSIF_Request or SIF_ServiceInput operation. In the case of a synchronous transport error, the SIF_Error subelements (including Error Category and Error Code) are mapped to equivalent subelements in the SOAP v1.1 Fault element. The SIF WSDL specifies that SOAP Fault is to be returned synchronously (on the same HTTP/S connection) in the **fault** part of every SIF WSDL operation whenever a transport level error

	occurs. In the case of an asynchronous user level error, the SIF_Error element (renamed “UserLevelError”), is returned, contained in the SOAPBody.
SIF_Status	SIF_Status spans the cases where: 1. The synchronous response to an earlier “Get” operation is being returned. There is an internal SIF_Data subelement. 2. The synchronous transport status of any operation other than a just issued “Get” is being reported. There is no internal SIF_Data subelement. 3. An asynchronous Status operation is issued to report the successful arrival of the synchronous Status response to an earlier Get operation (case 2 above). There is no internal SIF_Data subelement. Each is expanded below.
SIF_Status with SIF_Data (AgentACL, ZoneStatus, or SOAP-ENV)	The SIF_Status with SIF_Data is removed. In this case, SIF_Code and SIF_Desc have been “promoted” to the SOAP Header under the names “GetReturnCode” and “GetReturnDescription”. SIF_Data(empty except for its choice of subelements) has also been removed. One of its three subelements is now the **output** returned synchronously (via the same HTTP/S connection) upon the success of the **input** part of the corresponding WSDL “Get” operation. This is the only time a Status message is not returned. Specifically: • AgentACL is in the SOAP Body in response to GetAgentACL • ZoneStatus is in the SOAP Body in response to GetZoneStatus • SOAP-ENV containing the SOAP Header and SOAP Body of the next message for a Pull Mode Agent, is in the SOAP Body in response to that agent issuing a GetMessage.
SIF_Status without SIF_Data (Status)	It includes SIF_Code and SIF_Desc. This is the top level SOAP Body element for the **output** “part” of almost every SIF WSDL operation whenever the **input** part was successfully received. It is also sent asynchronously by an SWA to indicate successful reception of an AgentACL, ZoneStatus or SOAP_Env response to a Get operation.
SIF_Event (Event)	SIF_Eventhas been removed. The object name and object action are both in the SIF_Header. The SOAP Body contains “Event” which is an xs:any.
SIF_Request (Query) (ExtendedQuery)	SIF_Requesthas been removed. Query and Extended Query are the new top level elements. Max Buffer Size and Version (of the response) are both in the SIF_Header.

SIF_Response (QueryResults) (ExtendedQueryResults)	SIF_Responsehas been removed. ObjectData (renamed QueryResults) and ExtendedQueryResults are the new top level elements. RequestMsgId and all packet information is contained in the SIF_Header.
SIF_SystemControl	SIF_SystemControlhas been removed. The choices under SIF_SystemControlData (Ping, Sleep, Wakeup, GetMessage,GetZoneStatus, GetAgentACL, CancelRequests and CancelServiceInputs) are the new top level elements contained in the SOAP Body.
Zone Status Operations (SIF_ServiceInput, SIF_ServiceOutput, SIF_ServiceNotify)	All three Zone Status operations have been removed. All elements above SIF_Body are now in the SOAP Header. There is a ServiceHeader element under SIF_Header which contains ServiceName, ServiceOperation and ServiceMsgId. The SOAP Body for all service operations will consist of a single child element (named ServiceInput, ServiceOutput or ServiceNotify) whose contents are equivalent to SIF_Body (xs:Any). This allows any Zone Service message to be carried by the SOAP transport.

Example:
The following XML instance fragment shows the entire SIF_Header within a SOAP Header, for a Change Event Operation on the StudentPersonal Object, being invoked on the Student Admin SIF Web Application in the Acme Middle School Zone.

```
<SIF_Headerxmlns=”http://www.sifassociation.org/transport/soap/2.x”soap:mustUnderstand=”1”>
<TransportVersion>2.5</TransportVersion>
<DataModel> http://www.sifassociation.org/us/datamodel/2.x</DataModel>
<DataModelVersion>2.5</DataModelVersion>
<ZoneId>AcmeMiddleSchool1.CoyoteDistrict.Arizona.US</ZoneId>
<MsgId>A3E90785EFDA330DACB00785EFDA330E</MsgId>
<SIF_Security>
<SIF_SecureChannel>
        <SIF_AuthenticationLevel>3</SIF_AuthenticationLevel>
<SIF_EncryptionLevel>4</SIF_EncryptionLevel>
</SIF_SecureChannel>
</SIF_Security>
<Timestamp>2010-10-24T15:58:33.984Z</Timestamp>
<SourceId>AcmeZIS</SourceId>
<DestinationId>StudentAdmin</DestinationId>
<MsgName>Event</MsgName>
<EventAction>Change</EventAction>
<ObjectName>StudentPersonal</ObjectName>
</SIF_Header>
```

Q.3.3WS-Addressing elements within the SIF Message SOAP Header

All SIF SOAP messages will support WS-Addressing. This is not directly reflected in the supplied WSDL, as it requires WS-Policy assertions that may not be supported on every platform. Support for including these elements in the SOAP Header would normally be provided by the Web Service development platform the service is implemented on.

The following table indicates how the values for these elements must be assigned. In terms of WS-Addressing, whenever an End Point Reference (EPR) is indicated, only its“address”subelement is mandatory.

WS-Addressing Element	**Char**	**Usage in SIF Message SOAP Header**
wsa:To	C	URL of destination. This element is required when the issuer of aSIF message initiates an HTTP/S connection. It must be omitted (or set to the WS-Addressing “anonymous” setting of *http://www.w3.org/2005/08/addressing/anonymous)* when a Status or Error response is being sent back synchronously over the same connection. For all Agentinitiated messages exchanges this MUST be set to the ZIS URL. For all ZIS initiated message exchangesit MUST be the URL provided by the Agent at Registration time (contained within the matching SIF_Protocol/SIF_Propertysubelement in theSIF_Register message).
wsa:From	O	Endpoint Reference of Source. The following MUST be true when this element is present. For the ZIS, the “Address” subelement within the EndpointReference will be the URL for the ZIS. For all Agent issued messages, the “Address” subelement within the From EndpointReferencewill be determined by one of the following conditions: **1. An Agent is issuing a client Request to the ZIS.** In this case, the URL provided at Agent Registration time (contained within the SIF_Protocol subelement in the SIF_Register message)is used. An example is when the Agent issues a Provision message. This is true for all SOAP messages issued by a Pull Mode Agent. **2. A PushMode Agent is responding to a Service Request.** In this case the URL provided by the Agent will match the URL that the Service Request arrived on.
wsa:MessageID	O	The value is unique for all messages in the Zone, and when present, it must beequivalent to SIF_MsgId.
wsa:Action	M	This defines the SOAP Body Contents, and indicates the operation being requested. It supersedes the value of the soapAction HTTP attribute. If that value is present, it must agree with the value in wsa:Action. The rules for constructing this AttributedURI are taken from the WS-Addressing standard conventions for WSDL1.1. The names of each input and output operation are included directly in the SIF WSDL. This requires that theAction valuesMUST

be:

Input: {target namespace}/{port type name}/{input name}
Output: {target namespace}/{port type name}/{output name}
Fault: {target namespace}/{port type name}/{operation name}/Fault/{fault name}

Examples (from Zone WSDL):

Provision Request from Agent to ZIS <In>:
http://www.sifassociation.org/contract/Zone-S11/2.x/Provision
Status Response from ZIS: <Out>:
http://www.sifassociation.org/contract/PushAgent-S11/2.x/ProvisionStatus
Error Response from ZIS:<Fault>
http://www.sifassociation.org/ contract/PushAgent-S11/2.x/Fault/ProvisionError

Because of the asynchronous nature of SIF data exchanges there is no Output message defined for a SIF WSDL operation other than an acknowledgement (whether Status or Error). For example, a SIF_Response message is sent as the Input of a new Response operation (triggered by the earlier reception of a Request message), rather than as the Output of the Request operation. The same is true for the relationship between ServiceInput and ServiceOutput.

Statusmessages (equivalent to the SIF_Status subelement in SIF_Ack) are the Output component of every defined SIF operation.

There are two varieties of possible successful responses to any SIF Operation. They are:

- **A Status with a SIF_Data subelement:** There are 4 operations where the Status within the Ackcontains additional data.
    - SIF_Register (Output returns Agent ACL permission)
    - SIF_GetZoneStatus(Output Returns Zone Status)
    - SIF_GetAgentAcl (Output returns Agent ACL permission)
    - SIF_GetMessage (Output returns next message to a Pull Agent)
- **A Status without a SIF_Data subelement.**This is the successful Output message to every other SIF Service Operation.

Error messages (equivalent to the SIF_Error subelement in SIF_Ack) are defined as the Fault part of every operation. Their Action value is indicated above.

It is also possible that a SOAP Fault may be sent for transport or encoding errors at a lower level of the stack. For these situations, the value of wsa:Action is predefined as:

http://www.w3.org/2005/08/addressing/soap/fault

It is also possible that a WS-Addressing Fault may be sent where the destination is unreachable or where the requested service action was not supported. For these

		situations, the value of wsa:Action is predefined as: http://www.w3.org/2005/08/addressing/ fault
wsa:RelatesTo	C	SIF_MsgId of related message (GUID). When it is present it has the values: • SIF_OriginalMsgId in SIF_Ack (used in Status and Error messages) • SIF_RequestMsgId in SIF_Response (Used in Query and Extended Query Response messages) • SIF_ServiceMsgId in SIF_ServiceOutput (Used in ServiceOutput)
wsa:ReplyTo	O	This is the recipient of the Status message for a successful operation. It SHOULD generally be omitted, but when present it MUST have the value "http://www.w3.org/2005/08/addressing/anonymous" This is the "anonymous URI" value which indicates that there is no real endpoint available for this address. Therefore any reply will be sent back in the HTTP/S response message (i.e. synchronously). Any other value would cause the HTTP/S connection to be broken and a new one established to the URI specified here (i.e. an asynchronous message pattern).
wsa:FaultTo	O	This is the recipient of the Error message for a failed operation. It SHOULD generally be omitted, but when present it MUST have the value "http://www.w3.org/2005/08/addressing/anonymous" This is the "anonymous URI" value which indicates that there is no real endpoint available for this address. Therefore any error will be sent back in the HTTP/S response message (i.e. synchronously). Any other value would cause the HTTP/S connection to be broken and a new one established to the URI specified here (i.e. an asynchronous message pattern).

Q.3.4Illustrated Example

The complete XML instance of theSOAP Message conveying a StudentPersonal Change Event (update phone number) to the Student Admin SIF Web Application in the Acme Middle School Zone (from the ZIS)is shown below. The WSDL operation is “Event”.

```
<soap:Envelopexmlns:soap=”http://schemas.xmlsoap.org/soap/envelope”>
<soap:Headerxmlns:wsa=”http://www.w3.org/2005/08/addressing”>
<wsa:To>https://AcmeHost:443/StudentAdmin</wsa:To>
<wsa:From>
<wsa:Address>https://AcmeHost:443/ZoneControl</wsa:Address>
</wsa:From>
<wsa:MessageID>A3E90785EFDA330DACB00785EFDA330E</wsa:MessageID>
<wsa:Action>http://www.sifassociation.org/message/soap/2.x/PushAgent-S11/Event</wsa:Action>
<wsa:ReplyTo>
<wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
</wsa:ReplyTo>
<wsa:FaultTo>
<wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
</wsa:FaultTo>

<SIF_Headerxmlns=”http://www.sifassociation.org/transport/soap/2.x” soap:mustUnderstand=”1”>
<TransportVersion>2.5</TransportVersion>
<DataModel> http://www.sifassociation.org/us/datamodel/2.x</DataModel>
<DataModelVersion>2.5</DataModelVersion>
<ZoneId>AcmeMiddleSchool1.CoyoteDistrict.Arizona.US</ZoneId>
<MsgId>A3E90785EFDA330DACB00785EFDA330E</MsgId>
<SIF_Security>
<SIF_SecureChannel>
        <SIF_AuthenticationLevel>3</SIF_AuthenticationLevel>
<SIF_EncryptionLevel>4</SIF_EncryptionLevel>
</SIF_SecureChannel>
</SIF_Security>
<Timestamp>2010-10-24T15:58:33.984Z</Timestamp>
<SourceId>AcmeZIS</SourceId>
<DestinationId>StudentAdmin</DestinationId>
<MsgName>Event</MsgName>
<EventAction>Change</EventAction>
<ObjectName>StudentPersonal</ObjectName>
</SIF_Header>
</soap:Header>

<soap:Body>
<Eventxmlns=”http://www.sifassociation.org/messages/soap/2.x”>
<StudentPersonal
xmlns=”http://www.sifassociation.org/datamodel/us/2.5”
RefId="D3E34B359D75101A8C3D00AA001A1652">
<PhoneNumberList>
<PhoneNumber Type="0096"><Number>(312) 555-1234</Number></PhoneNumber>
</PhoneNumberList>
</StudentPersonal>
</Event>
</soap:Body>
```

```
</soap:Envelope>
```

Notes

- The SOAP Header consists of global WS-Addressing elements and a SIF_Header complex element. The SIF_Header element “must be understood”, so if a non-SIF application receives a SIF SOAP message, it will throw a SOAP Fault.

- The value of wsa:Toused by the ZIS matches the URL contained in the DataModel Property of the SIF_Protocol message in the original Register message sent by the receiving Agent.

- The value of wsa:action will be the same for all SIF Events, no matter in what Zone that event message is issued. It is a URL consisting of “namespace/port type/input name” where port type is “PushAgent-S11” and operation in this case is “Event”. This will be true for all other message types as well.

- The SIF_Headeris a child element of the SOAP Header, and maintains its “SIF_” Prefix, but none of its subelements in the SOAP Header do, as they are scoped by a unique name space.

- The SOAP Body of thisSIF message contains the single“Event” child elementwhich does not have a “SIF_” prefix and is of type xs:Any. All of its subelements will be defined by the Data Model schema of the SIF objects being conveyed. This is true for all SIF messages carried in the SOAP Body.

- In this example, the namespace for the Data Model is "**http://www.sifassociation.org/usdatamodel/2.5**"

Q.3.5Transport Errors

The SOAP transport has a different set of errors than the SIF_ACK/SIF_Error mechanism, and a completely different way of representing them (for example strings instead of numeric codes are used).WS-Addressing then maps these SOAP faults into the SOAP Body in a very specific way.

This subsection defines the error message/error code mapping between the two transports. It ensures that error conditions can be reported and understood between SWAs and HTTPS agent/application pairs,

Q.3.5.1 SOAP1.1 Faults

SOAP errors are conveyed through “SOAP Faults”. These messages always have the following wsa:Action value:

```
<wsa:Action>http://schemas.xmlsoap.org/ws/2004/08/addressing/fault<wsa:Action>
```

The SOAP Body contains a top level SOAP Fault element consisting of a set of subelements, only the following of which are important to the SIF error mapping:

- Faultcode: Only one of 4 possible values (see table below)
- Faultstring: A description of the problem (always equated to SIF_Error/SIF_Desc)
- Detail: Application specific error informationwhich MUST be included if the fault was caused by the contents of the SOAP Body of the message being reported, but MUST NOT be included otherwise.

There are 4 "categories of SOAP v1.1 errors. Each is described in the table below. Note that conversion to SIF Error category and Error Code is implicit in 3 of them.

SOAP Error Category	Meaning	SIF Equivalent	Try Again?
VersionMismatch	Invalid Namespace in SOAP Element (ex: SOAP v1.2 message received)	Faultstring contains the reason. The implicit SIF Error values are: • Category 1 (XML Validation) • Code 2 (Message not well formed)	No
MustUnderstand	An immediate child of the SOAP Header with a "must understand attribute" was not understood. SIF_Header has such an attribute and most likely one part of this was not understood	Faultstring contains the reason. The implicit SIF Error values are: • Category 1 (XML Validation) • Code 3 (Generic Validation)	No
Server	Message could not be processed for reasons other than Message Contents	Faultstring contains the reason. The implicit SIF Error values are: • Category 10 (Transport Error) • Code 4 (Unable to establish Communication) In this case, resending might be a correct strategy after a suitable delay (the behavior is unchanged when receiving the equivalent SIF Error).	Maybe
Client	Message did not contain appropriate information.	This class of SOAP fault carries SIF Error information (Category, Code, Description and Extended Description) in the SOAP Body.	No

Q3.5.2Mapping SIF elements to the SOAP Fault Header

The following table formalizes the above example of how to set up the element values in a SOAP Fault message, which is sent from an Agent when it rejects a message, back to the ZIS which forwarded it originally.

SOAP Fault (Agent to ZIS)	Value of the SOAP Fault element in the body, if it exists
WS-Addressing Elements	
wsa:To	The URL of the ZIS
wsa:From	The URL of the Agent
wsa:MessageId	A unique message ID for this Fault
wsa:Action	The fixed string for all SOAP 1.1 Fault messages
wsa:RelatesTo	The Message ID of the message being rejected

<SIF_HeaderElements>	
SIF_Timestamp	The time of Error message creation
SIF_SourceId	The Source ID of the Agent
SIF_OriginalSourceId	The Source ID of the Agent which originally posted the message being rejected, and **not** the ZIS. It corresponds to the SIF_OriginalSourceId element in SIF_Ack.

Q.3.6Message-specific Mapping Issues

There are several "special cases" involved in mapping the HTTP/HTTPS message schemas to their SOAP transport equivalents that are examined in greater depth here.

Q.3.6.1 Initialization of a SIF Web Application

SIF_Register is the first message issued by the SWA, and the first indication to the ZIS that the SWA exists, and will be using the SOAP transport. There are two specific requirements placed on the SWA issuing this message.

- The Registration message MAYbe directed to the HTTPS port (443), the SOAP port (880) or any other port supported by the ZIS.
- The underlying transport for this message MUST be the HTTPS secure equivalent version of HTTP v1.1

The individual SIF_Protocol subelements within the Registration message from SIF applications using the SOAP transport, are REQUIRED to be in accordance with the following rules.

SIF_Protocol Component	Char	Type	Meaning
Type	M	Attribute	The range of legal enumerated values in DefinedProtocolsType have been extended to include "HTTP-SOAP1.1" and "HTTPS-SOAP1.1". One of these settings MUST be contained in the Type attribute.
Secure	M	Attribute	Unchanged in meaning Set to "Yes" or "No" depending upon whether the protocol provides a secure channel.
SIF_URL	C	Element	For Push Mode Agents, this is the HTTP/S URL subsequently used by the ZIS to invoke operations on the Web Service provided by that agent.
SIF_Property	M	Container	Thereis a Property Name/Value pair defining the SOAP Mapping Version number.

The following SIF_Propertypairs are defined for this release.

SIF_Name	Char	SIF_Value for this release
TransportVersion	M	"2.5"
DataModelVersion	M	"2.5"

Q.3.6.2A SWA Pull Mode Agent issues a GetMessage

If the GetMessageoperation is successful, this MUST result in a complete SIF Message being “packaged” within the returning SOAP Body. To make this understandable to the SWA, the internal message data must be packaged exactly as if it came from another SWA, whether or not this was actually the case.

ZIS MUST convert every HTTP/HTTPS message into its SOAP equivalent before placing it into the queue for the SWA Pull Agent. In other words, what was a SIF_Message element has to be converted into a SOAP Envelope, containing a SOAP Header (with equivalent WS-Addressing and SIF_Header parts) and a SOAP Body (with the message data).

The following table contrasts the top level element mapping between one of the SIF Response message packets being returned to a Pull Mode Agent, sent across each of the supported transports. Note that in the case of the SOAP transport, this is a QueryResults message, which is a renamed ObjectData element (see next section).

HTTPS Transport	SOAP Transport
SIF_Message	SOAP Envelope
SIF_Ack	SOAP Header
SIF_Header	WS-Addressing elements
SIF_Status	SIF_Header (for Status)
SIF_Code& Description	Return Code and Description
SIF_Data	SOAP Body
SIF_Message	SOAP Envelope
SIF_Response	SOAP Header
SIF_Header	WS-Addressing elements
Packet control elements	SIF_Header (for QueryResults)
	Packet control elements
SIF_ObjectData	SOAP Body
	QueryResults

Here the SOAP Body for the Status message contains both the SOAP Header and SOAP Body of the QueryResults messageit conveys (i.e. the equivalent of a SIF_Message).It is not necessary to bring the Packet Control elements of the nested QueryResults up to the actual SOAP Header for this message because:

- When sent in response to a series of GetMessages, the individual QueryResults packets will have different Message Ids. This meets the SOAP requirement for unique Message Ids

- The total size of the SOAP Message will reflect the Maximum Buffer size in the same way the total size of the SIF_Ack does over the HTTP/HTTPS transport.

- The packetization information makes sense only when the internal QueryResultsmessage is “unpacked”. Therefore it can remain in the inner SIF_Header element without having to be promoted to the SIF_Header for the Status message.

The actual exchange sequence in this case exactly parallels the existing one. Assuming there are no errors:

- Pull Agent issues a GetMessage to the ZIS.<In>

- ZIS synchronously responds (on the same HTTP/S connection) with a "data full" Status message to the Agent, containing the SOAP envelope for the next message in the Agent's Queue, in the SOAP Body of the Status. <Out>
- Pull Agent issues an "Immediate" Status message to the ZIS confirming successful reception of the new SOAP message (on a new HTTP connection). The ZIS can now free up the message in the Agent's Queue.<In>
- ZIS synchronously responds with a Status message indicating whether or not additional messages are in the Pull Agent queue.<Out>

Note that there are no "Intermediate" or "Final" Status messages exchanged in this sequence. Selective Message Blocking (SMB) is not supported over the SOAP transport.

Q.3.6.2 A User-level Error Occurs

There are several asynchronous "request / response" message exchange sequences supported over the SOAP transport:

- Query / QueryResponse
- ExtendedQuery / ExtendedQueryResponse
- ServiceInput / ServiceOutput

In all these cases, the message exchange orchestration is identical. The table below illustrates it for a Query / Query Response:

Initiating Agent	ZIS	Object Provider
Invoke Query →	→ Receive Query	
Receive Status ←	←Send Success Status to Agent	
	Forward Query to Object Provider →	→Receive Query
	Receive Status ←	← Send Success Status to ZIS
		<Process Query and create response>
	Receive QueryResponse←	←Send QueryResponse back to ZIS
	Send Success Status to Provider →	→Receive Status
Receive QueryResponse←	←ForwardQueryResponse to Originating Agent	
Send Success Status to ZIS →	→Receive Status	

In all cases, the Status message is sent back synchronously (on the same HTTP/HTTPS connection as the original message arrived on). If a transport error occurred, a SOAP Fault (with the Error Category, Error Code, and Error String contained in the SOAP Fault) is returned instead.

However there is the additional case where a user level error occurs, causing the object provider to reject the QueryRequest, and issue an asynchronous Error message instead of a QueryResponse. The Error message is then returned asynchronously exactly as the QueryResponse would have been.

Such asynchronous user errors are the only errors reported using the equivalent of the SIF_Error schema, and the Error operation must therefore be supported by both the PushAgent and ZIS web services.

Q.4 ZIS/SWA Functionality Mapping to WSDLInterface

A web application relies on the relevant Web Service Description Language (WSDL) documents to define the interfaces of the services it may access as a web client,and the operations it must implement as a web service.

The SIF standard defines the complete set of messages the ZIS will accept and send to connected agents (and their applications), along with the order in which these messages must be invoked (choreography). For example, a SIF agent/application must first Register itself in the Zone, before it can Provision itself as a subscriber toEvents subsequently posted for one or more object types (via the Event operation). Only after both operations compete successfully will the application actually receive Event messages. This exact message set and choreography is enforced for SIF Web Applications.

The remainder of this subsection describes the set ofWSDL v1.1 files which determine the interface to:

- The SIF Infrastructure Web Service (SIWS) provided by the ZIS for accessing its messaging functionality. The operations provided by the interface (Port Type) to this service areinvokedover the SOAP transport by SIF Web Applications, which act as web clients when they do so.

- Push Mode SIF Web Applications (PushSWAs) provide aWSDL End Point URL at Registration time, whereitsdefined Server Interface is located. This URL is used to inform the ZIS of where to initiate an HTTP/S connection when a SWA operation (ex: Event, Query) must be invoked. For these messages, the PushSWA essentially acts as the Web Service, and the ZIS plays the role of the Web Client.

Pull Mode SIF Web Applications (PullSWAs) always initiate message exchanges with the ZIS by invoking the "GetMessage" operation. Since they in effect act as pure web clients, they require no corresponding WSDL.

PushMode Agents must provide responses for all invocations on every operation defined in the PushAgent WSDL, even if they don't support them. Examples would be an incoming Query reaching a non-object provider or an incoming ServiceIn invoked on an application that was not a Zone Service. Where such operation support is not present, the Push Mode agent MUST respond by sending a SOAP message with either:

- A SIF Error message with an error category of 12 ("Generic Message Handling") and an error code of 2 ("Message not supported")
- A SOAP Fault with a faultstring value corresponding to a WS-Addressing Error "ActionNotSupported".

The ZIS MUST respond to and provide support for every operation defined in the ZoneWSDL.

Q.4.1 WSDL Overview

The following principles are reflected in the design of the WSDL files for both the ZIS and the PushSWA.

Q.4.1.1WSDL File Structure and Granularity

A given WSDL v1.1 file defines the complete external interface to a Web Service. There is one such WSDL file (Zone.wsdl) which MUST be supported by ZIS, and a second (PushModeAgent.wsdl) which MUST be supported by all Push Mode SIF Web Applications.

Both files import a file which defines the full set of WSDL messages both the ZIS and Push Agents share (Common.wsdl). They containa single Port Type (interface), which supports the group ofrelated operations defined below. Each of these operationsroughly corresponds to a supported SIF_Message type in the HTTP/S infrastructure. Each operation is "bound" to the SOAP 1.1 transport over HTTP/S in the supplied WSDLs.

Both files have a single WSDL Service entry with an address set to the WSDL Anonymous URI. An agent developer or Zone integrator need only replace this with the URL of the actual ZIS-provided service for the physical Zone before supplying it to the Agent as a complete WSDL description of the service it will be accessing.

In the case of a ZIS, the Push Mode agents will supply this URL at Registration time. The Anonymous URI should be set to any value that will allow a dynamic determination of the URL for each of the multiple agents that are simultaneously registered in the zone.

Q.4.2 WSDL Versioning and Namespaces

The version of both the ZIS and Push Agent WSDL files is defined as follows:

The specific major and minor release numbers of the SIF messaging schema will be contained in the "documentation" subelement within "definitions".

```
<documentation>Version 2.5</documentation>
```

The "compatibility" version number will also be appended to the back of the WSDL target namespace URL.

For the XML schema files that define both the ZIS and Push Mode Agent WSDLs, the following Namespace definitions were used:

- Default Namespace: http://schemas.xmlsoap.org/wsdl/
- Target Namespace: http://www.sifassociation.org/messages/soap/2.x

Q.4.3WSDL Operations

There is roughly a 1:1 correlation between the types of SIF messages defined for the HTTP/S transport, and the WSDL operations defined within the Port Types of both the ZIS and Push Agent web services. The WSDL operation name is located in the SOAP Header in element wsa:Action. All the data corresponding to that operation is located under the single child element of the SOAP Body.

Q.4.3.1 Legal WSDL Operations

The following table defines the legal WSDL operations that comprise the Port Type interfaces of both the SIF Infrastructure Web Service (SWIS) – i.e. the ZIS, and Pull Mode Agent web services. In several cases a single SIF Message Type maps to multiple WSDL operations. The reverse is never true.

All WSDL operations are defined for the Zone Web Service and MUST be supported by the ZIS. WSDL Operations with "(*)" are defined for all Push Mode Agent Web Services as well.

SIF Message Component	WSDL Operation	Comment
SIF_Ack/SIF_Status	Status	Issued to the SIWS (ZIS) only. Also serves as the synchronous output method for every other operation (see Note below)
SIF_Ack/SIF_Error SIF_Response/SIF_Error SIF_ServiceNotify/SIF_Error	Error	SIF_Error Schema. In case of transport errors, the elements are mapped to SOAP Fault.
SIF_Event/SIF_ObjectData/SIF_EventObject	Event(*)	Object Name and Object Action moved to SIF_Header
SIF_Provide	Provide	
SIF_Provision	Provision	
SIF_Register	Register	
SIF_Request/SIF_Query	Query(*)	Was Request subtype now separate operation
SIF/Request/SIF_ExtendedQuery	ExtendedQuery(*)	Was Request subtype now separate operation
SIF_Response/ObjectData	QueryResults(*)	Was Response subtype now separate operation **Note: Name changed for clarity**
SIF_Response/ExtendedQueryResults	ExtendedQueryResults(*)	Was Response subtype now separate operation
SIF_Subscribe	Subscribe	
SIF_SystemControl/SIF_SystemControlData		
/SIF_Ping	Ping (*)	Empty contents
/SIF_Sleep	Sleep (*)	Empty contents
/SIF_Wakeup	Wakeup (*)	Empty contents
/SIF_GetMessage	GetMessage	Issued by Pull Mode Agent (pure web client) only. Synchronous Status response (on same HTTP/S connection) will contain complete SOAP Envelope of next messagewithin its SOAP Body.
/SIF_GetZoneStatus	GetZoneStatus	ZoneStatus object data is returned in the synchronous ZoneStatus response
/SIF_GetAgentACL	GetAgentACL	AgentACL data is returned in the synchronous AgentACL response
/SIF_CancelRequests	CancelRequests(*)	Internal list of Message IDs is not replicated in SOAP Header.
/SIF_CancelServiceInputs	CancelServiceInputs(*)	Internal list of ServiceMessage IDs is not replicated in SOAP Header.
Synchronous Response Messages		
	SOAP_Envelope	Synchronous successful response to GetMessage
	ZoneStatus	Synchronous successful response to GetZoneStatus
	AgentACL	Synchronous successful response to

		GetAgentACL
	SOAP_Env	Synchronous successful response to GetMessage
SIF_Unprovide	Unprovide	
SIF_UnRegister	Unregister	Empty contents.
SIF_Unsubscribe	Unsubscribe	
SIF_ServiceInput/SIF_Body	ServiceInput (*)	Contents of ServiceInput element in SOAP Body are xs:Any
SIF_ServiceInput/SIF_Error	SOAP Fault	
SIF_ServiceOutput/SIF_Body	ServiceOutput (*)	Contents of ServiceOutput element in SOAP Body are xs:Any
SIF_ServiceOutput/SIF_Error	SOAP Fault	
SIF_ServiceNotify/SIF_Body	ServiceNotify (*)	Contents of ServiceNotify element in SOAP Body are xs:Any

Note: The "Status" message is special in several ways.

- It MUST be the synchronous output method for every SIF operation that is successful

It is also can be invoked as an input method itself. After a client receives a "data full" status to an input request (containing a Message, AgentACL or ZoneStatus), it must in turn invoke either the Status operation on the ZIS to indicate success, or send the ZIS a SOAP Fault to indicate an error.

Q.4.4SIF Infrastructure Web Service (SIWS) Groups and Associated Operations

The operations are divided into the following "groups". All group operations are supported in a single WSDL Port interface of a single WSDL Service, implemented by the ZIS. The web client invoking the set of operations contained in these groups MUST be an Agent.

Grouping	***Operations***
Zone Provisioning	*Register, Unregister, Provide, Unprovide, Subscribe, Provision, GetZoneStatus, GetAgentACL, Status*
Zone Data Model	*Event, Query, ExtendedQuery,QueryResults, ExtendedQueryResults, CancelRequests*
Zone Service	*ServiceNotify, ServiceInput, ServiceOutput, CancelServiceInputs*
Zone Queue	*GetMessage*
Zone System Control	*Ping, Sleep, Wakeup*

Q.4.4.1 Zone Provisioning Group

This SIF Infrastructure Web Service (SIWS) operationgroup encapsulates the ZIS operations which support an application provisioning itself as a user and supplier of Zone resources. Thesecan beinvoked by both PushMode and PullMode SIF Web Applications (SWAs).

The Register operation is where the PushMode SWA MUST provide its WSDL End Point URLs in the SIF_Protocol element. This is used by the ZIS (operating in web client mode) to invoke the SWA as a Web Service, so asynchronous messages (Events, Queries, ServiceInputs) can be delivered.

Zone Management (status and security) information can also be obtained through the operations in this group.

Q.4.4.2 Zone Data Model Group

The operations contained in the Data Model groupare used by a SIF Web Applicationto post Events, make Query and Extended Query requests, and (in the case of Object Providers) to supply Query and Extended Query responses.

They encapsulate the full (and identical) set of CRUD operations the ZISsupports for access to all Object Data. The ZIS MUST use content based routing on the supplied object type (in the SOAP Header) to forwardthe operation to any/alllegal SIF components in the Zone, whether they are an HTTP/HTTPS Agent or SWA.

Q.4.4.3 Zone ServiceGroup

This group allows Zone Service Clients toinvoke Zone Service operations over the SOAP transport, and allows a Zone Service to be written as a SIF Web Application.

The operations of the Zone Service group map directly to the three message types first introduced in the SIF v2.4 infrastructure expansionin support of Zone Services. The ServiceInput operation should only be invoked by client applications of one or more of the defined SIF Zone Services (Assessment, Student Record Exchange, Student or Staff Identifier). The ServiceNotify and ServiceOutput operations should only be invoked by a Zone Service SWA.

The operations in this WSDL group thus“homogenize” the interface to all Zone Services in the same way the interface to all Object Providers conforms to the “CRUD” model. Any SIF Zone Service(of whatever variety) can only be accessed through operations in this Web Service group at the message level. The operation(and arguments)specific to an individual service invocation will be embedded in the SOAP Body, rather than in the WS-Addressing “Action”.

Defining and externalizing unique WSDL definitions for each Zone Service type is a future enhancement to the SIF standard.

Q.4.4.4 Zone Queue Group

The GetMessage operationis used exclusively by Pull Mode SIF Web Applications. It allows them to function solely as web clients, by supportingan operation which “gets the next message to be delivered”. This operation SHOULD be invoked synchronously by these clients only when they are ready to receive a new message, and it eliminates the need for them to provide a call back URL at Registration time, or support any SIF-provided WSDL.

Pull Mode Agents must respond with a Status message on success, or a SOAP Fault on error. In terms of WSDL Operations (and assuming success):

- The Push Mode SWA invokes the GetMessage operation (the Input) and receives from the ZIS the Status message containing the next message in the SWA queue (the Output)

- As discussed earlier, the Push Mode SWA MUST then asynchronously invoke the Status operation of the ZIS to acknowledge successful reception of the “gotten” message. It receives a synchronous Status message back on the same HTTP/S connection, which indicates whether the ZIS has any additional messages in the Pull Agent’s queue.

- ZIS support for Synchronous Message Blocking (SMB) is not guaranteed for Push Mode SWAs.

Q.4.4.5 Zone System Control Group

The three operations in this Group provide the SWA with network level functionality, including the ability to sleep, wake up, and test (ping) whether its ZIS partner is active.

Q.4.5 Push Mode SIF Web Application (PushSWA) Groups and Associated Operations

As noted earlier, a Pull Mode SIF Web Application (SWA) MUST use the GetMessage operation of the Zone Queue Interface (see above) to request its messages in sequential fashion.

The Push Mode SWA support for incoming asynchronous messagesis mapped to three“groups”. All group operations are supported in a single WSDL Port interface of a single WSDL Service. In all cases, the web client invoking the set of operations contained in these groupsMUST be the ZIS.

Group Type	Char	*Operations*
PushSWA Data Model	O	*Event, Query, ExtendedQuery,QueryResults, ExtendedQueryResultsCancelRequests*
PushSWA Zone Service	O	*ServiceNotify, ServiceInput, ServiceOutput, CancelServiceInputs*
PushSWA System Control	M	*Ping, Sleep, Wakeup,*

Q.4.5.1 Push SWA Data Model Group

The operations contained in theData Model groupare used by the ZIS to issue(using content based routing) posted Events to Subscribers, Queries and ExtendedQuery responses to Object Providers, and Query and ExtendedQuery results back to requesters of object data.

They are all shared with the ZIS (both the ZIS and Push Mode Agents can receive them) and their message components are defined in the Common.wsdl file.

Q.4.5.2 PushSWA Zone Service Group

As with the ZIS WSDL, the operations in the Zone Service group map directly to the message types added by the SIF v2.4 infrastructure expansion (ServiceInput, ServiceOutput, and ServiceNotify).

A Push Mode SWA which is a Zone Service or a client of a Zone Service MUST be able receive these messages asynchronously, at the WSDL End Point specified during its registration. Messages corresponding to the operations of this group are also defined in the Common.wsdl file.

Q.4.5.3Push SWA System Control Group

Supporting the three operations of this group provide the ZIS with network level functionality, including the ability to sleep, wake up, and test (ping) whether its SWA partner is active. Support for all of these operations by the SWA is mandatory. Messages corresponding to the operations of this group are also defined in the Common.wsdl file.

Q.5 HTTP/S Interoperability

The Basic Profile 1.2 mandates a set of requirements on HTTP/S running under SOAP, some of which vary from the way SIF uses HTTP/S directly as the transport protocol. These differences are summarized in the following table and wherever possible SHOULD be visible only to the ZIS, which MUST bridge the following differences.

In most cases the necessary conversions are straightforward. Where they are not, they are expanded.

Situation/Protocol Aspect	**Existing SIF HTTP/S Transport**	**BP1.2 - HTTP/S under SOAP**
Successful Reception of Message	Only a 200 “OK” HTTP status code	Either a 200 “OK” or 202 “Accepted” HTTP status code
HTTP Errors (non-SOAP)	The SIF HTTPS protocol uses the 200-OK response notice to communicate all responses If a client receives any 3xx, 4xx, or 5xx response notices, it must treat these responses as if a transport error has occurred.	"400 Bad Request" HTTP status code, if a HTTP Request message is malformed "405 Method not Allowed" HTTP status code if a HTTP Request message's method is not "POST" 415 Unsupported Media Type" HTTP status code if a HTTP Request message's Content-Type header field-value is not permitted by its WSDL description.
SOAP Errors	The SIF HTTPS protocol uses the 200-OK response notice to communicate all responses If a client receives any 3xx, 4xx, or 5xx response notices, it must treat these responses as if a transport error has occurred.	500 "Internal Server Error" code only SOAP Fault contained in SOAP Body, **<This requires a conversion between HTTP/S codes 200 and 500 when a SIF_Error is being converted to a SOAP Fault message.>**
HTTP Connection Strategies <Note: Since all applications are connected through the ZIS, the ZIS may support differing connection strategies depending upon the transport protocol used to communicate with the partner.>	Use persistent connections. The client may send additional POST requests and receive the HTTP responses using the same connection. Clients SHOULD use persistent connections for performance reasons but MUST be able to use non-persistent connections if the server does not wish to use persistent connections	An HTTP/S connection is guaranteed to persist across only a single WSDL operation (input and output message exchange). This exchange is thus “synchronous. Only the anonymous URI may be used as the address for wsa:ReplyTo and wsa:FaultTo to ensure a synchronous response is sent back to the wsa:From endpoint on the existing HTTP/S connection.

Character Encoding	UTF-8	UTF-8
Message Compression/Encoding	It is RECOMMENDED that servers return a 406 (Not Acceptable) status when a requested encoding cannot be negotiated. If a 406 is received, the client SHOULD assume compression using the specified algorithm(s) is not supported and retry communication as per SIF HTTPS Transport or SIF HTTP Transport It is RECOMMENDED that servers unable to process a particular content encoding return a 415 (Unsupported Media Type) status code.	The set of content-encodings allowed by HTTP is open-ended and any besides 'gzip', 'compress', or 'deflate' are an extensibility point. There are no fixed error codes reserved for compression/encoding errors. MTOM is not supported in this release.
HTTP soapAction Attribute	N/A	When present, this value must agree with the value of the wsa:Action element in the SOAP Header.