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).

However because Zone Services support a richer interface than CRUD, the change was visible to its clients. . 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 existing partners of the original agent / application.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.

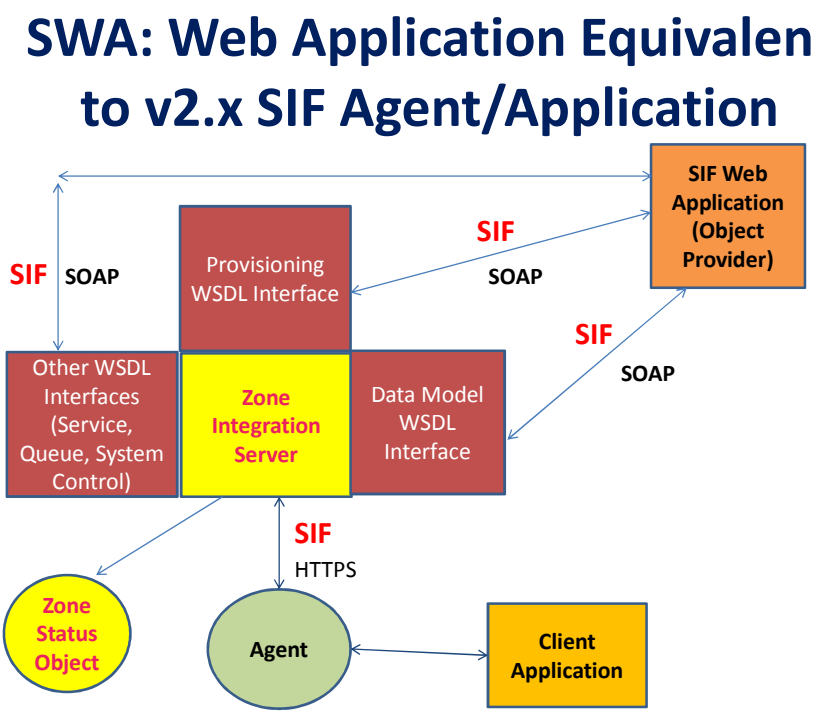

Figure 1: Example of the SIF Infrastructure Web Service(SIWS) transparently and simultaneously supportingboth a“Pull Mode” SIF Web Application (SWA)over SOAP, and an “original” SIF Agent / Application component over the original HTTPS transport.

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 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 theSIF_Messagesubschemas (ex: SIF_Response) is provided when mapping to the SOAP transport.

Each existing HTTP/HTTPS transport messageschema appears in modified form in its SOAP equivalent. This was done for the following reasons:

- The SOAP Header describes what type of message is being conveyed, and the SOAP Body conveys it. There was no need for a single top level SIF_Message container and it 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) positioned within the SOAP Header. While this will not always prevent the ZIS from needing to examine the SOAP Body before routing the message (for example when supporting XML Filtering requirements, or

where payload version conversion needs to be performed), itshould do so in the majority of cases and it presents a much more "SOAP-natural" mapping for thedata being exchanged.

- The SIF_Ack message has been broken up into its component pieces (becoming either a SIF_Status or a SIF_Error). The former message is the Output component of many SIF Service Operations and the latter message is reported as a SOAP Fault.
- It allowed the new SOAP message schemas to be made independent of the objects they convey. This enables them to be used in SIF solutions from different locales (US, UK, AU) despite the differences in the types of objects these locales support.

The XML Namespace for the entire set of message schemas carried within the SOAP Body in this release is: http://www.sifassociation.org/messages/soap/2.x. **These schemas retain the "SIF_" prefix convention for subelements.**

Review issues.

- Should SOAPmessage schema namespace behttp://www.sifinfo.org/infrastructure/soap/2.x instead?
- Should new message schemas for SOAP Transport (carried in SOAP Body) still have SIF_ prefixes?

The XML Namespace for the SIF elements contained in the SOAP Header for this release is: http://www.sifassociation.org/transport/soap/1.0. Except for the SIF_Header element itself, all other SIF-specific subelements within the SOAP Header will not have the "SIF_" prefix.

The following table contrasts the top level element mapping between two SIF Response messages being sent across each of the supported transports.

HTTPS Transport	SOAP Transport
SIF_Message SIF_Response SIF_Header Packet control elements SIF_Response Data elements	SOAP Header WS-Addressing elements SIF_Header (specific to SOAP transport) Message Type Packet control elements SOAP Body Response SIF_Response Data elements

All data elements unique to a specific object are contained in the SOAP Body. As a result, the SOAP Header is independent of the object being conveyed.

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". This contains the following two type of information:

- General SIF messaging information required by the ZIS to route a SIF message.
- Message type-specific information which is also needed for routing and other decisions

Note: Any elements identified after the phrase "equivalent to" in the tables below refer to the original HTTPS Transport SIF Message schemas.In those cases wherea SIF element has been relocated tothe SOAP Header, it has been removed from the equivalent SOAP Transport SIF Message schema.

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 elements existed within the original SIF_Message schema hierarchy they have been removed in their SOAP Transport equivalents.

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. Set to "1.0" for this release.
LocaleType	O	Locale for the Data message. Current values are "US", "UK" and "AU", but the list is extensible.The complete list is contained in Appendix A, Common Types.
DataModelVersion	O	Version of the Data Model which defines the schema for the set of XML elements contained in the SOAP Body payload of this message. Equivalent to the SIF_Version attribute contained in SIF_Message. Set to "2.x" for this release.
ZoneId	M	URI which uniquely identifies the Zone containing both the Sender and Receiver of this SOAP message. By convention it will be "." delimited, starting with most specific identification such as school and working rightward to identify the higher levels. (Ex*: AcmeMiddleSchool1.CoyoteDistrict.Arizona*) The initial (most specific) field SHOULD be identical to the SIF_ZoneId attribute returned in SIF_ZoneStatus (Ex: *AcmeMiddleSchool1*)
Security	O	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. Equivalent to the SIF_Security element in SIF_Header, with the SIF_ prefixes removed from itself and all subelements.

TimeStamp	M	Time of message creation. Equivalent to the SIF_TimeStamp element in SIF_Header
MessageType	M	The unique value for this SIF Message type (ex: "Event" for SIF_Event messages). The complete set of values is defined in "MessageNameType", which is documented in the Common Type Appendix A.
SourceId	M	Equivalent to the SIF_SourceId value in the 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_DestinationId in the SIF_Header. It is used by the ZIS tocontent-route the message based upon matching it with a pre-stored SIF_SourceId. For SWAs it is set only if the message is "Directed".
Contexts		The list of Contexts to which the message applies. Currently only the default value is defined. Equivalent to the SIF_Contexts value in the SIF_Header.
PacketInfo	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.
PacketInfo/SOAPMessageID	M	The SOAP MsgId for the first packet of this message. If this is the first packet, this value MUST duplicate wsa:MessageId
PacketInfo/Number	M	xs:positive integer corresponding to the packet number
PacketInfo/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 an Event Operation being invoked on the Student Admin SIF Web Application in the Acme Middle School Zone.

<SIF_Headerxmlns="**http://www.sifassociation.org/transport/soap/1.0**">
<TransportVersion>**1.0**</TransportVersion>

```
<LocaleType>US</LocaleType>
<DataModelVersion>2.x</DataModelVersion>
<ZoneId> AcmeMiddleSchool1.CoyoteDistrict.Arizona.US</ZoneId>
<Security>
<SecureChannel>
        <AuthenticationLevel>3</AuthenticationLevel>
<EncryptionLevel>4</EncryptionLevel>
</SecureChannel>
</Security>
<TimeStamp>2010-10-24T15:58:33.984Z</TimeStamp>
<MessageType>Event</MessageType>
<SourceId>AcmeZIS</SourceId>
<DestinationId>StudentAdmin</DestinationId>
```

Q.3.2.2Message-specific Information

Each of these elements may be contained in the SOAP Header,depending upon what type of SIF message is being conveyed in the SOAP Body.They are not universally defined for all SIF SOAP messages.

Where equivalent elements existed within the original SIF_Message schema hierarchy, they have not been removed in their SOAP Transport equivalents, to ensure the SOAP Body remains fully self-descriptive.

SIF_Message Type	New SIF_Header subelements contained in the SOAP Header	Char	Usage / Meaning / Equivalent in HTTPS Transport Message Schema
SIF_Ack	OriginalSourceId	C	Equivalent to the SIF_OriginalSourceId element in SIF_Ack if the message type being conveyed is anAck Note: The wsa:RelatesTo element elsewhere in the SOAP Header is equivalent to the SIF_OriginalMsgId element in SIF_Ack
SIF_Event	EventObjectName	C	Equivalent to the ObjectName attribute in SIF_Event if the message being conveyed is an Event.
SIF_Event	EventAction	C	Equivalent to the Action Attribute (“Add”, “Delete”, “Change”) of SIF_Event if the message being conveyed is an Event.
SIF_Request	RequestObjectName	C	Equivalent to the ObjectName attribute of SIF_QueryObject, found in SIF_Request, if the

			message being conveyed is a Request. Note that while there are multiple ObjectName attributes in SIF_ExtendedQuery, they will remain in the SOAP SIF_ExtendedQueryschema and not be represented here. If needed for routing, the ZIS will go into the SOAP Body to retrieve them.
SIF_ServiceInput SIF_ServiceNotify	Service	C	The name of the Zone Service (Ex: *"serviceAgencyStudentIdManagement"*) Equivalent to the SIF_Service element in SIF_ServiceInput and SIF_ServiceNotify, if the message being conveyed is a ServiceInput or ServiceNotify.
SIF_ServiceInput SIF_ServiceNotify	Operation	C	The name of the Zone Service Operation being invoked. Ex: *"ResolveIdentifier"*) Equivalent to the SIF_Operation element in the SIF_ServiceInput and SIF_ServiceNotify if the message being conveyed is a ServiceInput or a ServiceNotify.
SIF_ServiceInput SIF_ServiceNotify SIF_ServiceOutput	ServiceMsgId	C	The unique Id of this service request invocation. Equivalent to the SIF_ServiceMsgId element in SIF_ServiceInput and SIF_ServiceNotify, if the message being conveyed is a ServiceInput or ServiceNotify. In the case of a SIF_ServiceOutput message, the ServiceMsgId element will be the value of the wsa:RelatesTo element in the SOAP Header. Note that while there is also a list of SIF_ServiceMsgId's in SIF_CancelService, they will remain in the SOAP CancelService schema and not be represented in the SOAP Header. If needed, the ZIS will have to go into the SOAP Body to retrieve them

Notes: The following SIF_Messagesubschema elements are not duplicated in the SOAP Header.

1. SIF_MaxBufferSize

While used by SIF_Register, SIF_Request, and SIF_ServiceInput, this information does not need to be carried in the SOAP Header as it does not affect delivery of this message. If needed it can be pulled from its current location and used by the ZIS.

2. SIF_RequestMessageID
This element is carried in wsa:RelatesTo for all SIF_Response equivalent SOAP messages.

While there is a list of SIF_RequestMsgId's in SIF_CancelRequest, they will remain in the SOAP CancelRequest schema and not be represented in the SOAP Header. If needed, the ZIS MUST go into the SOAP Body to retrieve them.

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. In this case, both the Object Name and Event Type information will also be contained in the Event schema within the SOAP Body.

```
<SIF_Headerxmlns="http://www.sifassociation.org/transport/soap/1.0">
<TransportVersion>1.0</TransportVersion>
<LocaleType>US</LocaleType>
<DataModelVersion>2.x</DataModelVersion>
<ZoneId> AcmeMiddleSchool1.CoyoteDistrict.Arizona.US</ZoneId>
<Security>
<SecureChannel>
        <AuthenticationLevel>3<AuthenticationLevel>
<EncryptionLevel>4<EncryptionLevel>
<SecureChannel>
<Security>
<TimeStamp>2010-10-24T15:58:33.984Z</TimeStamp>
<MessageType>Event</MessageType>
<SourceId>AcmeZIS</SourceId>
<DestinationId>StudentAdmin</DestinationId>
<EventObjectName>StudentPersonal</EventObjectName>
<EventObjectAction>Change</EventObjectAction>
</SIF_Header>
```

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

In terms of WS-Addressing, where an End Point Reference (EPR) is indicated, only the Address element is mandatory.

WS-Addressing Element	Char	Usage in SIF Message SOAP Header
wsa:To	M	URL of destination. For all Agent issued messages this MUST be set to the ZIS URL. For the ZIS, with one exception, it MUST be the URL provided by the Agent at

		Registration time that matches the endpoint of the corresponding Port Type (contained within the matching SIF_Protocol/SIF_Propertysubelement in theSIF_Register message). Depending upon the wsa:Action it MUST match the URL property value for the corresponding property Port Type name: • DataModel • ZoneServices • SystemControl The exception is when the ZIS is sending an Ack message (status or error) in response to a previously received message. In this case, the URL will match the SIF_Protocol/SIF_URL element provided at Agent Registration time.
wsa:From	M	Endpoint Reference of Source. 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 EndpointReferencewill be determined by one of the following conditions: **1. An Agent is issuing a client Request to one of the ZIS Port Types.** In this case, the URL provided at Agent Registration time (contained within the SIF_Protocol subelement in the SIF_Register message)MUST be 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 on one of its own Port Types.** In this case the URL provided by the Agent MUST match the URL that the Service Request arrived on.
wsa:MessageID	M	Equivalent to SIF_MsgId
wsa:Action	M	The rules for constructing this AttributedURI are taken from the WS-Addressing standard. Neither the wsa:Action nor 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} / {operation name}Request Output: {target namespace} / {port type name} / {operation name}Response Examples: Provision Request from Agent

		http://www.sifassociation.org/transport/soap/1.0/datamodel/provisionRequest Status Response from ZIS: http://www.sifassociation.org/transport/soap/1.0/datamodel/provisionResponse 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. **Status**messages (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 SIF_Status with a SIF_Data subelement:** There are 4 operations where the Status within the Ackcontains additional data. o SIF_Register (Output returns Agent ACL permission) o SIF_GetZoneStatus(Output Returns Zone Status) o SIF_GetAgentAcl (Output returns Agent ACL permission) o SIF_GetMessage (Output returns next message to a Pull Agent) • **A SIF_Status without a SIF_Datasubelement** This is the successful Output message to every other SIF Service Operation. **Error** messages (equivalent to the SIF_Error subelement in SIF_Ack) are reported to the wsa:FaultTo addressing element within the SOAP Header of the Input component of the Operation. They are returned as SOAP Faults.
wsa:RelatesTo	O	SIF_MsgId of related message (GUID). It has the values: • SIF_OriginalMsgId in SIF_Ack • SIF_RequestMsgId in SIF_Response • SIF_ServiceMsgId in SIF_ServiceOutput
wsa:ReplyTo	O	End Point Reference to notify on success. This is the recipient of the Status message. It defaults to wsa:From
wsa:FaultTo	O	End Point Reference to notify of failure. This is the recipient of the Error message. It defaults to wsa:From

Q.3.4Illustrated Example

The complete XML instance of being invoked on the Student Admin SIF Web Application in the Acme Middle School Zone.

The complete XML instance of theSOAP Message conveying a StudentPersonal Change Event to the Student Admin SIF Web Application in the Acme Middle School Zoneis shown below.

```
<soap:Envelopexmlns:soap=”http://www.w3.org/2003/05/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:Action>http://www.sifassociation.org/transport/soap/1.0/datamodel/eventRequest</wsa:Action>
<wsa:ReplyTo>
<wsa:Address>https://AcmeHost:443/ZoneControl</wsa:Address>
</wsa:ReplyTo>
<wsa:FaultTo>
<wsa:Address>https://AcmeHost:443/ZoneControl</wsa:Address>
</wsa:FaultTo>
<wsa:MessageID>A3E90785EFDA330DACB00785EFDA330E</wsa:MessageID>

<SIF_Headerxmlns=”http://www.sifassociation.org/transport/soap/1.0”>
<TransportVersion>1.0</TransportVersion>
<LocaleType>US</LocaleType>
<DataModelVersion>2.x</DataModelVersion>
<ZoneId> AcmeMiddleSchool1.CoyoteDistrict.Arizona.US</ZoneId>
<Security>
<SecureChannel>
        <AuthenticationLevel>3</AuthenticationLevel>
<EncryptionLevel>4</EncryptionLevel>
</SecureChannel>
</Security>
<TimeStamp>2010-10-24T15:58:33.984Z</TimeStamp>
<MessageType>Event</MessageType>
<SourceId>AcmeZIS</SourceId>
<DestinationId>StudentAdmin</DestinationId>

<EventObjectName>StudentPersonal</EventObjectName>
<EventObjectAction>Change</EventObjectAction>
</SIF_Header>
</soap:Header>
```

```
<soap:Body>
<Eventxmlns=”http://www.sifassociation.org/messages/soap/2.x.”>
<SIF_ObjectData>
<SIF_EventObject>
<StudentPersonalRefId="D3E34B359D75101A8C3D00AA001A1652">
<PhoneNumberList>
<PhoneNumber Type="0096"><Number>(312) 555-1234</Number></PhoneNumber>
</PhoneNumberList>
</StudentPersonal>
</SIF_EventObject>
</SIF_ObjectData>
</SIF_Event>
</soap:Body>

</soap:Envelope>
```

Notes

- The SOAP Header consists of global WS-Addressing elements and a SIF_Header complex element.
- The value of wsa:To matches the URL given 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 message is issued. This will be true for all other message types as well.
- The SIF Header child element of the SOAP Header maintains its “SIF_” Prefix, but none of its subelements in the SOAP Header do, as they are scoped by a unique name space.
- All SIF message elements in the SOAP Body retain their SIF_ prefix, which keeps them aligned with the HTTP/HTTPS Transport schema.
- The first version of the SOAP Transport namespace is 1.0.

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 errors 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,

Q3.5.1 A non-SOAP SIF Error Message

The following example details how a typical error is reported.It illustrates a SIF_Ack/SIF_Error message being returned from a Library System to the ZIS, complaining about a SIS Event it received.

In this case an Authentication error has occurred because the Publisher’s certificate is not trusted. Here the SIF_Category value 3 indicates “Authentication error” as defined by its corresponding entry in the Error

Category table in Code Set Appendix B. The SIF_Code value 5 indicates the reason this occurred, as defined by its corresponding entry in the Authentication Error table also in Code Set Appendix B.

```
<SIF_Message Version="2.4" xmlns="http://www.sifinfo.org/infrastructure/2.x">
<SIF_Ack>
<SIF_Header>
<SIF_MsgId>CD5087FE3261545A31905937B265CE01</SIF_MsgId>
<SIF_Timestamp>2010-02-18T08:39:40-08:00</SIF_Timestamp>
<SIF_SourceId>RamseyLIB</SIF_SourceId>
</SIF_Header>
<SIF_OriginalSourceId>RamseySIS</SIF_OriginalSourceId>
<SIF_OriginalMsgId>1945CD783261545A31905937B265CE01</SIF_OriginalMsgId>
<SIF_Error>
<SIF_Category>3</SIF_Category>
<SIF_Code>5</SIF_Code>
<SIF_Desc>Sender's certificate is not trusted</SIF_Desc>
<SIF_ExtendedDesc>Agent requires certificate issued by ISD11 CA</SIF_ExtendedDesc>
</SIF_Error>
</SIF_Ack>
</SIF_Message>
```

Q3.5.2 A SOAP Transport SIF Fault Message

The following details the way the same SIF error would be reported over the SOAP v1.1 Transport if the Library System was a SIF Web Application.**Note that this XML instance document would be identical whether or not the original SIS system was also an SWA**.

```
<soap:Envelopexmlns:soap=”http://schemas.xmlsoap.org/soap/envelope”>
<soap:Headerxmlns:wsa=”http://www.w3.org/2005/08/addressing”>
<wsa:To>https://AcmeHost:443/ZoneControl</wsa:To>
<wsa:From>
<wsa:Address>https://AcmeHost:443/Library</wsa:Address>
</wsa:From>
<wsa:MessageID> CD5087FE3261545A31905937B265CE01</wsa:MessageID>
<wsa:Action>http://schemas.xmlsoap.org/ws/2004/03/addressing/fault</wsa:Action>
<wsa:RelatesTo> 1945CD783261545A31905937B265CE01</wsa:RelatesTo>

<SIF_Headerxmlns=”http://www.sifassociation.org/transport/soap/1.0”>
<TransportVersion>1.0</TransportVersion>
<LocaleType>US</LocaleType>
<DataModelVersion>2.x</DataModelVersion>
<ZoneId> AcmeMiddleSchool1.CoyoteDistrict.Arizona.US</ZoneId>
<Security>
<SecureChannel>
        <AuthenticationLevel>3<AuthenticationLevel>
<EncryptionLevel>4<EncryptionLevel>
</SecureChannel>
</Security>
```

```
<TimeStamp>2010-10-24T15:58:33.984Z</TimeStamp>
<MessageType>Error</MessageType>
<SourceId>RamseyLib</SourceId>
<DestinationId>AcmeZIS</DestinationId>
<OriginalSourceId>RamseySIS</OriginalSourceId>
</SIF_Header>
</soap:Header>

<soap:Body>
<soap:Faultxmlns="http://schemas.xmlsoap.org/soap/envelope/".>
<soap:faultcode>soap:client.3.5</soap:faultcode>
<soap:faultstring>Sender's certificate is not trusted</soap:faultstring>
<soap:detailxmlns:sif=”http://www.sifassociation.org/messages/soap/2.x.”>
<sif:SIF_Desc>Agent requires certificate issued by ISD11 CA.</sif:SIF_Desc>
</soap:detail>
</soap:Fault>
</soap:Body>
```

Notes:

The WS-Addressing **Action** indicates a SOAP Addressing fault. It is not a SIF defined Action

The **SIF Message Type** is “Error”. However there is no equivalent to SIF_Error in the SOAP Body.

The SOAP Body consists of a top level SOAP Fault element. The SIF namespace is not declared at this level because

- The SOAP Fault is not a SIF Message
- In most cases there will be no SIF-specific elements
- No Namespace in a SOAP Fault may be unqualified (as the SIF namespace is in every other SOAP Body top element).

The **SOAP Fault Code** is defined as “soap:Client” (one of several predefined and required Codes for SOAP 1.1, postfixed with a qualifier (“.3.5” in this case), which represents the “SIF_Category.SIF_Code” of the original error message. **<Note: This extension to faultcodeusage is allowed under the SOAP 1.1 encoding rules for this element. It must be confirmed that all WS developer tools will allow applications to set the Fault Code in this way.>**

The **SOAP Fault String** is equivalent to the SIF_Desc element in SIF_Error.

It is only in the case where a SIF_ExtendedDesc element was contained in the original SIF_Error, that the optional **Soap Detail** element must be included. In this case, the SIF Namespace must be declared (as non-Default) as some SIF specific information must be reported. Since the SIF_Error Message does not exist, the SIF_Desc element of the Status message is used to contain the extended error.

Q3.5.3Mapping SOAP Fault Elements

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 elements of interest	Char	Value of the SOAP Fault element in the body
WS-Addressing Elements		
wsa:To	M	The URL of the ZIS
wsa:From	M	The URL of the Agent
wsa:MessageId	M	A unique ID for this Fault message
wsa:Action	M	The fixed string for all SOAP 1.1 Fault messages
wsa:RelatesTo	M	The Message ID of the message being rejected
<SIF_HeaderElements>		
SIF_TimeStamp	M	The time of Error message creation
MessageType	M	The fixed value of "Error" will be used for all SOAP Faults
SIF_SourceId	M	The Source ID of the Agent
SIF_OriginalSourceId	M	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.
SOAP Body		
Fault/faultcode	M	One of 4 SOAP 1.1 preset values (see below) followed by the numeric values of SIF_Error/SIF_Category and SIF_Error/SIF_Code delimited at each level by a "."
Fault/faultstring	M	The SIF_Desc string of SIF_Error
Fault/faultactor		Unused
Fault/detail	O	Uses the qualified SIF namespace. An optional subelement directly maps the string value of: SIF_Error/SIF_ExtendedDesc into the string element of Status/SIF_Desc

Q3.5.4SOAP Fault Code Equivalents

There is a set of SOAP transport errors which may be auto-generated by specific development platform software, and which have Fault/faultcode values which therefore cannot always be mapped into the predefined set of SIF Error / SIF Category codes before they are placed on the wire. The table below shows how these non-numeric values can be assigned to their SIF_Error equivalents upon reception.

SOAP Fault	Error	SIF_Error Equivalent
faultcode is "soap:VersionMismatch"	The SOAP receiver saw a namespace associated with the SOAP envelope that it does not recognize. If it is ever seen, it will be treated as an XML Validation error	Category: 1 Code: 1

	– Generic As this error was not detected in the SOAP body, no Fault/detail element is required	
faultcode is soap:MustUnderstand	An immediate child of the SOAP header had **mustUnderstand**set to true. The receiver of the message did not understand the header. If it is ever seen, it will be treated as an XML Validation Error – Generic Validation As this error was not detected in the SOAP body, no Fault/detail element is required	Category:1 Code:3
faultcode is soap:Client	Something about the way that the message was formatted or the data it contained was wrong. The client needs to fix its mistake before the message can be resent. The majority of all SIF Errors will map to this faultcode.	SIF_Category and SIF_Codemay be postfixed to faultcode. If not, use Error Category 14 (SOAP Client Error), Code 1 (Generic Error)
faultcode is soap:Server	An error happened at the server. Depending on the nature of the error, the client may be able to resend the exact same message to the server and see it processed.	Use Error Category 15 (SOAP Server Error), Code 1 (Generic Error)

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 MUST arrive over the HTTPS port (443) or the SOAP port (880)
- The underlying transport for this message MUST be the HTTPS secure equivalent version of HTTP v1.1**<This needs to be reviewed>**

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 values in DefinedProtocolsType has been extended to include "SOAP-HTTP" and "SOAP-HTTPS". 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 will be the https or http URL used by the ZIS for sending ACK messages.
SIF_Property	M	Container	There will be a Property Name / Value pair defining the SOAP Mapping Version number. If a Push Mode Agent is registering, there will also be a Property Name / Value pair for every Port Type in the Push Agent WSDL which is supported by this Agent.

The following SIF_Propertypairs are defined for this release. The individual Port Property values may or may not be the same.

SIF_Name	Char	SIF_Value
SoapMappingVersion	M	"2.5"
DataModelPort	O	The https or http URL for contacting the Service End Point at which the Push Mode Agent supports the Data Model Operations.
ZoneServicesPort	O	The https or http URL for contacting the Service End Point at which the Push Mode Agent supports the Zone Service Operations.
SystemControlPort	O	The https or http URL for contacting the Service End Point at which the Push Mode Agent supports the System Control Operations.

Q.3.6.2A SWA Pull Mode Agent issues a SIF_GetMessage

If the SIF_GetMessage is successful, this must result in a complete SIF Message being "packaged" within the returning SIF_Status. 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.

If necessary, the ZIS must convert every HTTP/HTTPS message into its SOAP equivalent before placing it into the queue for the SWA Pull Agent.

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.

HTTPS Transport	**SOAP Transport**
SIF_Message SIF_Ack SIF_Header SIF_Status SIF_Code SIF_Data SIF_Message SIF_Response SIF_Header Packet control elements Response Data elements	SOAP Header WS-Addressing elements SIF_Header Message Type (“Status”) SOAP Body Status SOAP Header (for SIF_Response) WS-Addressing elements SIF_Header Message Type (“Response”) **Packet control elements** SOAP Body Response Response Data elements

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

- When sent in response to a series of GetMessages, the individual response 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 Response message 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..

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. 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 two WSDL 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 interfaces (Port Types) of this service areinvokedover the SOAP transport by SIF Web Applications, which act as web clients when they do so. All Service interfaces MUST be mapped to the same WSDL End Point URL.

- Push Mode SIF Web Applications (PushSWAs) provide one or more WSDL End Point URLs at Registration time, one for each of the three available Server Interfaces which are supported. These URLs are used (in a manner exactly analogous to a Push Mode Agent) to inform the ZIS of where to initiate an HTML connection when a SWA operation (ex: Event, Request) must be invoked. It does this by mapping the requested operation to the URL corresponding to the interface supporting that operation. 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), like their Pull Mode Agent equivalents, always initiate message exchanges with the ZIS by invoking the “Get” (next) Message operation. Since they in effect act as pure web clients, they require no corresponding WSDL.

Q.4.1 WSDL Overview

The following principles are reflected in the design of the WSDL files for both the ZIS and the PushSWA. They each import the “infrastructure-soap” XSD file to complete their SOAP binding.

Q.4.1.1Granularity

A given WSDL v1.1 file defines the complete external interface to a Web Service. There is one such WSDL file (Zis.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 contain multiple Port Types (interfaces), each of which supports a group ofone or morerelated operations.Each of these operations corresponds to a supported message type in the HTTP/S infrastructure.

A ZIS MUST accept and support all operations on all defined interfaces in the ZIS WSDL. A Push SWA may support some or all of the interfaces defined in its WSDL, but it must accept all operations for each interface it does support, even if the response to an operation invocation is limited to posting a SOAP Fault with Error Category 14 (SOAP Client Error) and Code 2 (Operation Unsupported).

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.4.3SIF Infrastructure Web Service (SIWS) Port Types and Associated Operations

The ZIS functionality has been mapped to the following series of "Port Type" WSDL Service Interfaces

Port Type	*Operations*
Zone Provisioning	*Register, Unregister, Provide, Unprovide, Subscribe, Provision, GetZoneStatus, GetAgentACL*
Zone Data Model	*Event, Request, Response, CancelRequest*
Zone Service	*ServiceNotify, ServiceInput, ServiceOutput, CancelServiceInput*
Zone Queue	*GetMessage, Status*
Zone System Control	*Ping, Sleep, Wake*

Q.4.3.1 Zone Provisioning Interface

This SIWS interface 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.

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

Zone Management (status and security) information can also be obtained through this interface.

Q.4.3.2 Zone Data Model Interface

The Data Model interface is used by a SWA to post Events, make Requests, and (in the case of Object Providers) to supply a Response to a received Request for object data.Its operations encapsulate the full (and identical) set of CRUD operations the ZISsupports for access to all Object Data. Behind this interface, 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 HTTP/HTTPS Agents or SWAs.

Q.4.3.3 Zone Service Interface

This interface 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 Interface map directly to the three message types first introduced in the SIF v2.4 infrastructure expansionin support of Zone Services. The ServiceInput operation defined by this interfaceshould 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 the Zone Service SWAs.

This WSDL interface thus interface “homogenizes” the interface to all Zone Services in the same way the interface to all Object Providers conforms to the identical “CRUD” model. Any SIF Zone Service(of whatever variety) can only be accessed through this Web Service interface at the message level,viathe “ServiceNotify”, “ServiceInput” and “ServiceOutput” operations. 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 separate WSDL definitions for each Zone Service type is a future enhancement to the SIF standard.

Q.4.3.4 Zone Queue Interface

This SIWS Interface is used exclusively by Pull Mode SIF Web Applications. It allows them to function solely as web clients, by providingthe 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.

They 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 on this interface (the Input) and receives from the ZIS the Status message containing the next message in the SWA queue (the Output)

- The Push Mode SWA then invokes the Status operation on this interface (the Input) to acknowledge successful reception of that message. This is the only time the Status message will be sent as the Input of any Operation. In all other cases, it is sent as the Output.

Support for Synchronous Message Blocking (SMB) is not guaranteed for Push Mode SWAs.

Q.4.3.5 Zone System Control Interface

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

Q.4.4 Push Mode SIF Web Application (PushSWA) Port Types and Associated Operations

As noted earlier, a Pull Mode 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 "Port Type" WSDL Service Interfaces. In all cases, the web client invoking the set of operations encapsulated by these interfacesMUST be the ZIS.

Port Type	Char	*Operations*
PushSWA Data Model	O	*Event, Request, Response, CancelRequest*
PushSWA Zone Service	O	*ServiceNotify, ServiceInput, ServiceOutput, CancelServiceInput*
PushSWA System Control	M	*Ping, Sleep, Wakeup,*

Q.4.4.1 Push SWA Data Model Interface

Where supported by the SWA, this Data Model interface is used by the ZIS to forward (using content based routing) posted Events to Subscribers, Requests to Object Providers, and Responses to Requesters of object data.

Q.4.4.2 PushSWA Zone Service Interface

As with the ZIS WSDL, the operations of the Zone Service Interface map to the message types added by the SIF v2.4 infrastructure expansion.

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. A Pull Mode SWA MUST use the GetMessage operation of the Zone Queue Interface (see below) to request these messages in sequential fashion.

Q.4.4.3Push SWA System Control Interface

The three operations of this Interface 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. **<This needs to reviewed.>**

Q.5Further Work needed<this section will not appear in the Standard>

The following areas in this document need further investigation, review or testing

Q.5.1 Detailed Fault Code Mapping

It is not clear whether applications on a variety of development platforms can always determine their soap:Server and soap:Client fault values. If not, the platform specific faultstring values may be placed on the wire, which means they must be interpreted after they arrive rather than before they are sent.

A table of these strings can be determined and documented so that the ZIS can map them to the appropriate SIF category / code values. An example is shown below:

WS-Addressing Fault	SIF_Error
wsa:DestinationUnreachable [Reason] "No route can be determined to reach the destination role defined by the WS-Addressing To"	Category 10 (Transport) Code 4 (Unable to establish connection).

Q5.2 HTTP Bridging

There may be potential differences between WS-I Basic Profile's use of HTTP and SIF HTTP which could affect the mapping. For example, when soap:Fault is issued, the HTTP status code may need to be 500.