SEMI E132.1-0310
SPECIFICATION FOR SOAP BINDING FOR EQUIPMENT CLIENT AUTHENTICATION AND AUTHORIZATION (ECA)

This Standard was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits & Reviews Subcommittee on December 16, 2009. Initially available at www.semi.org in February 2010. Originally published March 2005; previously published November 2008.

SUPERSEDED

Table of Contents

1 Purpose 5
2 Scope 5
3 Limitations 5
4 Referenced Standards and Documents 5
4.1 *SEMI Standards* 5
4.2 *ISO Standards* 5
4.3 *NIST Standards* 5
4.4 *OMG Standards* 5
4.5 *W3C Standards* 5
4.6 *Other Standards* 6
5 Terminology 6
5.1 *Abbreviations and Acronyms* 6
5.2 *Definitions* 6
6 Conventions 6
6.1 *Translating UML to XML Schema* 6
6.2 *Documenting XML Schema and WSDL Files* 7
6.3 *Documenting XML Schema with Diagrams* 8
6.4 *XML Schema Sample* 9
6.5 *Translating UML to WSDL* 10
7 SEMI E132 Authentication Mapping to SSL 11
7.1 *SSL Overview* 11
7.2 *Credentials and Digital Certificates* 12
7.3 *Enabling/Disabling SSL Authentication* 13
8 Mapping of SEMI E132 UML to XML Schema and WSDL 14
8.1 *WSDL Organization* 14
8.2 *SecurityAdmin Interface* 15
8.3 *SessionManager Interface* 35
8.4 *SessionClient Interface* 44
8.5 *InterfaceDiscovery Interface* 50
9 Message Transport 54

List of Figures

Figure 1 XML Schema Example Diagram 9
Figure 2 XML for Sample 10
Figure 3 SSL Basic Handshake Flow 12
Figure 4 SEMI E132.1 Security Credentials 12
Figure 5 XML Schema and WSDL File Organization 14
Figure 6 XML Schema and WSDL files for SEMI E132 15
Figure 7 Mapping the SecurityAdmin Interface to a WSDL portType 16
Figure 8 Sample SecurityAdmin WSDL Service Definition 18
Figure 9 UnauthorizedOperationType XML Schema Type 19
Figure 10 GetDefinedPrivilegesRequest XML Schema Element 20
Figure 11 GetDefinedPrivilegesResponse XML Schema Element 20
Figure 12 DefinedPrivilegesType XML Schema Type 21

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310

Figure 13 GetDefinedPrivilegesErrorType XML Schema Type 21
Figure 14 GetACLRequest XML Schema Element....22
Figure 15 GetACLResponse XML Schema Element 23
Figure 16 ACLEntrySelectorType XML Schema Type 23
Figure 17 PrivilegeAssignment XML Schema Type....24
Figure 18 RoleAssignment XML Schema Type....24
Figure 19 AddACLEntryRequest XML Schema Element....26
Figure 20 AddACLEntryResponse XML Schema Element 27
Figure 21 UnrecognizedPrivilegeType XML Schema Type 27
Figure 22 DeleteACLEntryRequest XML Schema Element 28
Figure 23 DeleteACLEntryResponse XML Schema Element....29
Figure 24 GetActiveSessionsRequest XML Schema Element 30
Figure 25 GetActiveSessionsResponse XML Schema Element....30
Figure 26 ActiveSession XML Schema Type 31
Figure 27 SetMaxSessionsRequest XML Schema Element 32
Figure 28 SetMaxSessionsResponse XML Schema Element....33
Figure 29 GetMaxSessionsRequest XML Schema Element....34
Figure 30 GetMaxSessionsResponse XML Schema Element 34
Figure 31 Sample SessionManager WSDL Service Definition 37
Figure 32 EstablishSessionRequest XML Schema Element....38
Figure 33 EstablishSessionResponse XML Schema Element 38
Figure 34 SessionPingRequest XML Schema Element....39
Figure 35 SessionPingResponse XML Schema Element 40
Figure 36 CloseSessionRequest XML Schema Type 41
Figure 37 CloseSessionResponse XML Schema Element....41
Figure 38 EnhancedEstablishSessionRequest XML Schema Element....42
Figure 39 EnhancedEstablishSessionResponse XML Schema Element....43
Figure 40 Sample SessionClient WSDL Service definition 47
Figure 41 SessionFrozenNotification XML Schema Element....48
Figure 42 SessionClosedNotification XML Schema Element....49

List of Tables

Table 1 Example Translation Table....7
Table 2 Example Translation Table for Operation Input/Output Arguments 7
Table 3 Example Schema/WSDL Document Description Table....8
Table 4 Altova XMLSPY Schema Diagram Symbols....8
Table 5 Example Interface WSDL Port Type Table....10
Table 6 Example Interface WSDL Binding Table....10
Table 7 Example Operation Binding Table 11
Table 8 Example PortType Operation Table 11
Table 9 XML Schema....15
Table 10 SecurityAdmin PortType Definitions 15
Table 11 SecurityAdmin Binding Definitions 16
Table 12 SecurityAdmin Port Type 17
Table 13 SecurityAdmin Binding....17
Table 14 SEMI E132 Error Codes....18
Table 15 Translation Table for UnauthorizedOperationError Class....19
Table 16 Translation Table for GetDefinedPrivileges Output Parameters 20
Table 17 Translation Table for DefinedPrivilege Class....20
Table 18 GetDefinedPrivileges Messages 21
Table 19 GetDefinedPrivileges PortType Operation....21
Table 20 GetDefinedPrivileges Operation Binding....22
Table 21 Translation Table for GetACL Output Parameters....22
Table 22 Translation Table for PrivilegeAssignment Class 24
Table 23 Translation Table for RoleAssignment Class 24
Table 24 Translation Table for Subject Class....25

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

Table 25 Translation Table for Privilege Class 25
Table 26 GetACL Messages 25
Table 27 GetACL PortType Operation 25
Table 28 GetACL Operation Binding 26
Table 29 Translation Table for AddACLEntry Input Parameters 26
Table 30 Translation Table for AddACLEntry Output Parameters 26
Table 31 Translation Table for UnrecognizedPrivilegeError Class 27
Table 32 AddACLEntry Messages 27
Table 33 AddACLEntry PortType Operation 28
Table 34 AddACLEntry Operation Binding 28
Table 35 Translation Table for DeleteACLEntry Input Parameters 28
Table 36 Translation Table for DeleteACLEntry Output Parameters 28
Table 37 DeleteACLEntry Messages 29
Table 38 DeleteACLEntry PortType Operation 29
Table 39 DeleteACLEntry Operation Binding 29
Table 40 Translation Table for GetActiveSessions Output Parameters 30
Table 41 Translation Table for ActiveSession Class 30
Table 42 Translation Table for HTTPEndPoint Class 31
Table 43 GetActiveSessions Messages 31
Table 44 GetActiveSessions PortType Operation 31
Table 45 GetActiveSessions Operation Binding 32
Table 46 Translation Table for SetMaxSessions Input Parameters 32
Table 47 Translation Table for SetMaxSessions Output Parameters 32
Table 48 SetMaxSessions Messages 33
Table 49 SetMaxSessions PortType Operation 33
Table 50 SetMaxSessions Operation Binding 33
Table 51 Translation Table for GetMaxSessions Output Parameters 34
Table 52 GetMaxSessions Messages 34
Table 53 GetMaxSessions PortType Operation 35
Table 54 GetMaxSessions Operation Binding 35
Table 55 XML Schema 35
Table 56 SessionManager PortType Definitions 35
Table 57 SessionManager Binding Definitions 36
Table 58 SessionManager Port Type 36
Table 59 SessionManager Binding 37
Table 60 Translation Table for EstablishSession Input Parameters 37
Table 61 Translation Table for EstablishSession Output Parameters 38
Table 62 EstablishSession Messages 38
Table 63 EstablishSession PortType Operation 39
Table 64 EstablishSession Operation Binding 39
Table 65 Translation Table for SessionPing Output Parameters 39
Table 66 SessionPing Messages 40
Table 67 SessionPing PortType Operation 40
Table 68 SessionPing Operation Binding 40
Table 69 Translation Table for CloseSession Input Parameters 41
Table 70 Translation Table for CloseSession Output Parameters 41
Table 71 CloseSession Messages 41
Table 72 CloseSession PortType Operation 42
Table 73 CloseSession Operation Binding 42
Table 74 Translation Table for EstablishSession Input Parameters 42
Table 75 Example of Equipment Capability Entry Values 43
Table 76 Translation Table for EstablishSession Output Parameters 43
Table 77 Translation Table for RequestRejectError Class 44
Table 78 EnhancedEstablishSession Messages 44
Table 79 EnhancedEstablishSession PortType Operation 44
Table 80 EnhancedEstablishSession Operation Binding 44

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

Table 81 XML Schema....45
Table 82 SessionClient PortType Definitions....45
Table 83 SessionClient Binding Definitions....45
Table 84 SessionClient Port Type....46
Table 85 SessionClient Binding....46
Table 86 Translation Table for SessionPing Output Parameters....47
Table 87 SessionPing Messages....48
Table 88 SessionPing PortType Operation....48
Table 89 SessionPing Operation Binding....48
Table 90 SessionFrozen Messages....48
Table 91 SessionFrozen PortType Operation....49
Table 92 SessionFrozen Operation Binding....49
Table 93 Translation Table for SessionClosed Input Parameters....49
Table 94 SessionClosed Messages....49
Table 95 SessionClosed PortType Operation....50
Table 96 SessionClosed Operation Binding....50
Table 97 InterfaceDiscovery XML Schema....50
Table 98 InterfaceDiscovery WSDL Definitions....50
Table 99 InterfaceDiscovery Operations....51
Table 100 Translation Table for EquipmentInterfaceDescription Parameters....52
Table 101 Translation Table for EquipmentInterfaceDescriptionType Parameters....52
Table 102 Translation Table for SessionGroupType Parameters....52
Table 103 Translation Table for InterfaceDescriptionType Parameters....53
Table 104 GetInterfaces Messages....53
Table 105 GetInterfaces PortType Operation....54

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

1 Purpose

1.1 The purpose of this specification is to provide an authentication implementation using SSL and a WSDL/SOAP implementation and XML Schema that corresponds to the UML model for Equipment Client Authentication and Authorization as defined by SEMI E132.

2 Scope

2.1 The scope of this document is the faithful representation of the ECA model in an XML schema and corresponding WSDL services. It will not add new domain information or concepts to the model. The only additions made are those needed to render useful WSDL or XML Schema.

NOTICE: SEMI Standards and Safety Guidelines do not purport to address all safety issues associated with their use. It is the responsibility of the users of the documents to establish appropriate safety and health practices, and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 None.

4 Referenced Standards and Documents

4.1 *SEMI Standards*

SEMI E121 — Guide for Style & Usage of XML for Semiconductor Manufacturing Applications

SEMI E128 — Specification for XML Message Structures

SEMI E132 — Specification for Equipment Client Authentication and Authorization

4.2 *ISO Standards*[1]

ISO/IEC 11578:1996 — Information Technology – Open Systems Interconnection – Remote Procedure Call (RPC)

4.3 *NIST Standards*[2]

NIST FIPS PUB 180-1 — Secure Hash Standard

4.4 *OMG Standards*[3]

OMG 01-09-67 — Unified Modeling Language, v1.4

4.5 *W3C Standards*[4]

Extensible Markup Language (XML) 1.0 (Fourth Edition) — W3C 16 August 2006; http://www.w3.org/TR/2006/REC-xml-20060816/

Namespaces in XML (Second Edition) — W3C 16 August 2006; http://www.w3.org/TR/2006/REC-xml-names-20060816/

XML Schema Part 0: Primer (Second Edition) — W3C, 28 October 2004; http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/

XML Schema Part 1: Structures (Second Edition) — W3C, 28 October 2004; http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[1] International Organization for Standardization, ISO Central Secretariat, 1 rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30; http://www.iso.ch

[2] National Institute of Standards and Technology, 100 Bureau Drive, Stop 3460, Gaithersburg, MD 20899-3460, USA. Telephone: 301.975.6478; http://www.nist.gov

[3] Object Management Group, Inc., 250 First Ave. Suite 100, Needham, MA 02494, USA; Telephone: 781.444.0404; Fax:781.444.0320; http://www.omg.org

[4] World Wide Web Consortium, Massachusetts Institute of Technology (MIT), Computer Science and Artificial Intelligence Laboratory (CSAIL), 32 Vassar Street Room 32-G515, Cambridge, MA 02139, USA; Telephone: 617.253.2613; Fax: 617.258.5999; http://www.w3.org

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310

XML Schema Part 2: Datatypes (Second Edition) — W3C, 28 October 2004; http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) — W3C, 27 April 2007; http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

SOAP Version 1.2 Part 2: Adjuncts (Second Edition) — W3C, 27 April 2007; http://www.w3.org/TR/2007/REC-soap12-part2-20070427/

Web Services Description Language (WSDL) 1.1 — W3C Note; http://www.w3.org/TR/wsdl

4.6 *Other Standards*

Secure Sockets Layer (SSL) Protocol 3.0; available from http://wp.netscape.com/eng/ssl3/draft302.txt

Web Services Interoperability (WS-I): Basic Profile Version 1.0a; http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.html

Internet Engineering Task Force (IETF) RFC 2459 — Internet X.509 Public Key Infrastructure Certificate and CRL Profile; http://www.ietf.org/rfc/rfc2459.txt

Internet Engineering Task Force (IETF) RFC 2616 — Hypertext Transfer Protocol – HTTP/1.1; http://www.ietf.org/rfc/rfc2616.txt

PKCS #12: Personal Information Exchange Syntax Standard, version 1.0; available from http://www.rsasecurity.com/rsalabs/pkcs/pkcs-12/

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *HTTP* — Hyper Text Transport Protocol

5.1.2 *SHA* — Secure Hash Algorithm

5.1.3 *SOAP* — Simple Object Access Protocol

5.1.4 *SSL* — Secure Sockets Layer

5.1.5 *UML* — Unified Modeling Language

5.1.6 *W3C* — World Wide Web Consortium

5.1.7 *WSDL* — Web Service Description Language

5.1.8 *XML* — eXtensible Markup Language

5.2 *Definitions*

5.2.1 *Unified Modeling Language (UML)* — a notation for representing object-oriented designs and views created by Booch, Rumbaugh, and Jacobson in order to merge their three popular notations plus aspects of other existing notations into a single object-oriented notation intended to be usable by all.

5.2.2 *eXtensible Markup Language (XML)* — a markup language used for representing data rich with context and content in documents and in communications. XML is an extension of SGML, a document-oriented markup language. It was created by W3C for use on the Internet.

6 Conventions

6.1 *Translating UML to XML Schema*

6.1.1 The reader is expected to have a working knowledge of the UML, XML, and Schema specifications (see ¶ 4.2 and ¶ 4.3). This document does not provide tutorial information on these subjects.

6.1.2 This document follows the guidelines for XML as outlined in SEMI E121.

6.1.3 Some of the key guidelines followed in this document are summarized here:

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

- Attributes of UML classes are generally represented as XML attributes, to improve efficiency in transferring metadata descriptions from the equipment. Exceptions to this convention are UML attributes that correspond to data structures or other constructs that cannot be represented as XML attributes, or may include text values that may need to include non-parsed character data (CDATA).
- Composition associations are mapped to a single XML element (or multiple such elements if the specified multiplicity is greater than 1).
- Aggregations are modeled as contained elements or arrays of references to one or more unique attributes of the target type.
- Inheritance is modeled using complexType extension. In cases where a reference to an abstract base class occurs in the UML model, a choice compositor for each of the possible derived types is used in order to restrict instance documents from including arbitrary extension through type substitution.

6.1.4 The translation of a UML class to XML Schema types is documented using a table format illustrated by Table 1. Operations defined for a UML class are translated into WSDL operations.

6.1.5 *Translation Table Column Header Description*

- *Attribute or Role Name* — If an attribute, the name of the attribute is placed here. If an association (including aggregation or composition), the role name from the UML diagram is placed here. Compositions are often not assigned roll names. In that case "none" is placed here.
- *UML Name/Type* — If an attribute, the data type of the UML attribute is place here. If an association, the type of association is placed here. The possible types are "Composition", "Aggregation", or the basic "Association". UML defines these three types of association.
- *XML Element or Attribute* — Lists the type of XML construct used to represent the UML attribute or association.
- *XML Name/Type* — Provides the name and data type of the resulting XML construct. The type may be a built-in type (e.g., xs:string), or a named type defined within the XML Schema.

Table 1 Example Translation Table

Attribute or Role Name	*UML Name/Type*	*XML Element or Attribute*	*XML Name/Type*
friend	association	element	Friend: HumanReferenceArray
employees	aggregation	element	Employees: HumanArray
family	composition	element	Family: HumanArray
name	string	element	Name: xs:string

6.1.6 *Translating Operations to XML Schema*

6.1.6.1 The translation of an operation defined for UML interface classes to XML Schema types is documented using a table format illustrated in Table 2. One row is provided for each possible input/output argument for the operation being described.

Table 2 Example Translation Table for Operation Input/Output Arguments

Argument Name	*Format*	*XML Element or Attribute*	*XML Name/Type*
<argument name from abstract specification>	<argument format from abstract specification>	<whether the argument is modeled as an Element or Attribute>	<namespace-qualified name of the attribute/element>

6.2 *Documenting XML Schema and WSDL Files*

6.2.1 Associated with this document are XML Schema and WSDL files that are core components of the specification. Each such document is described using a table format illustrated by Table 3.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310

Table 3 Example Schema/WSDL Document Description Table

File Name	<.xsd or .wsdl file name>
Target Namespace	<target namespace defined in the schema or WSDL file>
Imported/Referenced Namespaces	<namespaces imported or used by the schema or WSDL file>
Description	<brief description of the purpose/function of the schema or WSDL file>

6.3 *Documenting XML Schema with Diagrams*[5]

6.3.1 This document provides graphical representations of the included XML. Although no standard graphical notation for XML could be found, various XML tools have their own notation. This document will use the notation provided by XMLSpy from Altova Corporation. Figure 1 shows a sample XML diagram that will be used to provide a basis for explanation of the XML graphical notation used in the rest of the document.

6.3.2 In the diagram, rectangular boxes represent XML elements. The owner elements are on the left side while the elements contained within the owner element are on the right side. In the sample diagram, ParentType contains Child1, Child2, Child 3, and Child 4. In turn, Child2 contains Child2a, Child2b, and Child2c. The additional symbols (8-sided boxes) represent sequences or choices. See Table 4 for an explanation of these symbols.

Table 4 Altova XMLSPY Schema Diagram Symbols

	Denotes a required ordered sequence of the right hand elements with a cardinality of one for each element (sequence).
	Denotes an optional ordered sequence of the right hand elements with a cardinality of one for each element (sequence).
	Denotes a required, but unordered, sequence of the right hand elements with a cardinality of one for each element (all).
	Denotes a required choice of the right hand elements. Exactly one or two of the right hand elements must be present (choice).
	Denotes an element that contains parsed character data.
	Denotes a reference to a group or element defined elsewhere.

6.3.3 A graphic using a solid line is a required element; using a dashed line represents an optional element. Numbers or ranges in the lower right hand corner represent cardinality. The default cardinality is one.

[5] All images/graphics were created using Altova's XMLSPY®. Copyright 2003 Altova GmbH and reprinted with permission of Altova.

SUPERSEDED
Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

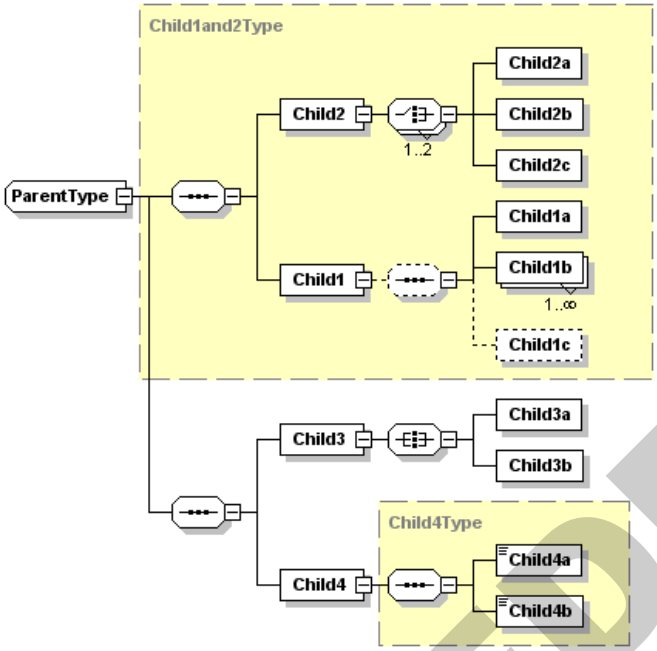

Figure 1
XML Schema Example Diagram

6.3.4 To simplify a diagram or help focus on a particular aspect, detail may be hidden. The 8-sided symbols have a small square on the right end. If a minus sign "–" is in the box, then all detail is shown. If the box contains a plus sign "+", then all detail to the right of that symbol is hidden. The example has no hidden detail.

6.3.5 The yellow (or grey if printed in monochrome) boxes indicate the use of other defined types. So, Child4 is of type "Child4Type". Child4Type defines Child4a and Child4b. This detail may be hidden in the diagram. Object oriented inheritance is typically represented in XML as type extension. In Figure 1, ParentType extends Child1and2Type by adding a sequence that includes Child3 and Child4.

6.3.6 Reading Figure 1 would yield the following additional information:

1. Child1and2Type is an ordered sequence of two items: Child2 and Child1.
2. Child1 contains an optional ordered sequence of Child1a, one or more Child1b, and (optionally) Child1c.
3. Child2 contains a choice of one or two of the following: Child2a, Child2b, and Child2c.
4. Child3 contains an unordered sequence of Child3a and Child3b.

6.4 *XML Schema Sample*

6.4.1 The sample XML Schema for the example shown in Figure 1, is presented below. Refer to the XML documentation referenced in ¶ 4.3 for a complete description of the syntax and semantics of XML Schema.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310

```
<xs:element name="Parent">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Child1"/>
            <xs:complexType>
               <xs:sequence minOccurs="0">
                  <xs:element name="Child1a"/>
                  <xs:element name="Child1b" maxOccurs="unbounded"/>
                  <xs:element name="Child1c" minOccurs="0"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="Child2">
            <xs:complexType>
               <xs:choice maxOccurs="2">
                  <xs:element name="Child2a"/>
                  <xs:element name="Child2b"/>
                  <xs:element name="Child2c"/>
               </xs:choice>
            </xs:complexType>
         </xs:element>
         <xs:element name="Child3">
            <xs:complexType>
               <xs:all>
                  <xs:element name="Child3a"/>
                  <xs:element name="Child3b"/>
               </xs:all>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Figure 2
XML for Sample

SUPERSEDED

6.5 *Translating UML to WSDL*

6.5.1 In general, UML interface classes defined in the abstract specification are translated into WSDL as portType definitions, and each portType definition has a corresponding WSDL binding definition. The following tables show the convention used for documenting the WSDL port type and binding definitions for a given UML interface class.

Table 5 Example Interface WSDL Port Type Table

Class Name	<UML interface name from abstract specification>
WSDL Port Type Name	<port type name used in WSDL>
SEMI E125 Operation → WSDL Operation	<operation name from UML interface> → <WSDL port type operation name>

Table 6 Example Interface WSDL Binding Table

SEMI E125 Class Name	<UML interface name from abstract specification>
WSDL Binding Name	<binding name used in WSDL>
SOAP Binding Style	<RPC or document>
SOAP Transport	<transport identifying URI>

6.5.2 Each operation defined for a given UML interface class has a corresponding operation definition and binding. Operations are described using a table format illustrated by Table 7 and Table 8.

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

Table 7 Example Operation Binding Table

SOAPAction	<the Content-Type header "action" value to be used for this operation>
Input Headers (WSDL Message, Required)	<the name of the WSDL message providing input headers, and whether or not the headers are required>
Output Headers (WSDL Message, Required)	<the name of the WSDL message providing output headers, and whether or not the headers are required>

Table 8 Example PortType Operation Table

Input Message Name	<WSDL message name used as input for the WSDL port type operation>
Output Message Name	<WSDL message name used as output for the WSDL port type operation>

7 SEMI E132 Authentication Mapping to SSL

7.1 *SSL Overview*

7.1.1 This section describes the use of SSL security protocol and supporting technology to implement the authentication scheme described in SEMI E132. SSL is application protocol independent and transparently supports any higher layered transport protocols such as SOAP over HTTP.

7.1.2 Figure 3 presents the basic message flow (handshake) of SSL to establish a secure connection and how it maps to the SEMI E132 authentication messages, please refer to the SSL documentation referenced in § 4.6 for complete specification on SSL. This specification defines several restrictions on the capabilities of SSL to the minimum necessary to support SEMI E132 authentication. These restrictions are described in more detail in the following section.

7.1.3 *SEMI E132 Restrictions on SSL*

7.1.3.1 X.509 v3 public key certificates as defined by RFC 2459 must be used to establish equipment and client identities, see X.509 documentation referenced in § 4.6 for complete details of X.509 structure and content. The certificate is sent when required during the SSL handshake.

7.1.3.2 Mutual authentication is required. Equipment shall always send its certificate and request for the client's certificate as part of the SSL handshake. Correspondingly, client will send its own certificate and the certificate verify message to equipment to allow client authentication.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310

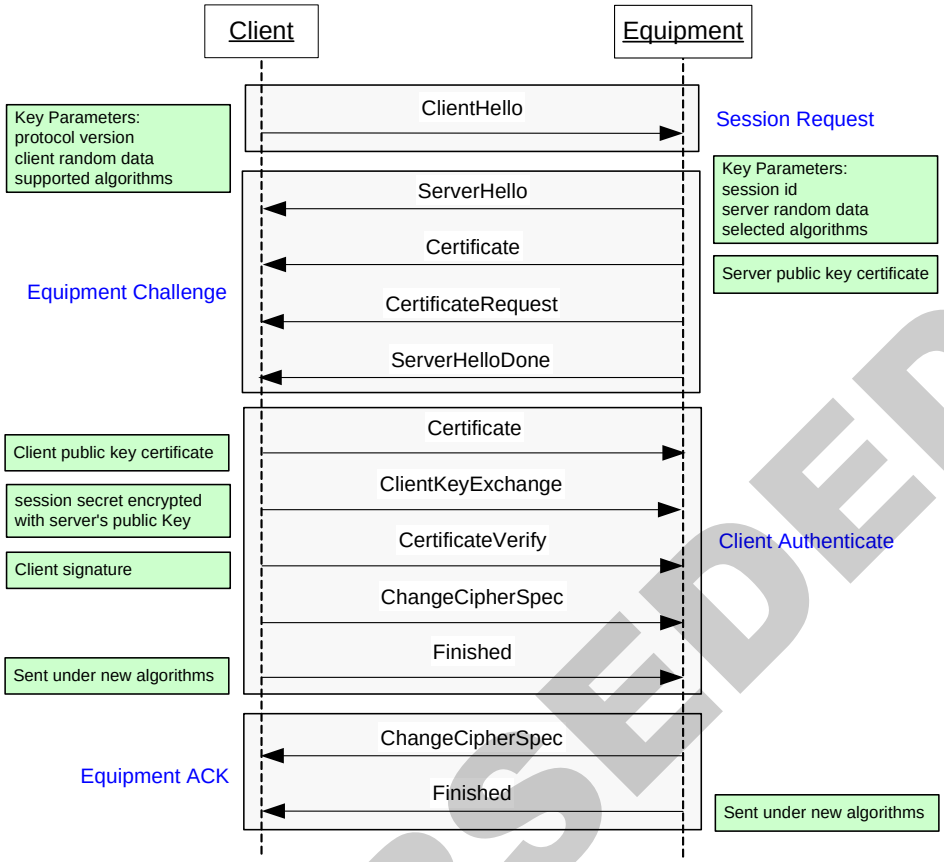

Figure 3
SSL Basic Handshake Flow

7.2 *Credentials and Digital Certificates*

7.2.1 *Overview*

7.2.1.1 X.509 certificates shall be used to establish identities for equipment and application clients. The certificate is used by SSL for authentication and to establish secure connections between equipment and clients. An illustration of the SEMI E132.1 security credentials showing role of certificates and ACL is presented in Figure 4. As shown, an X.509 certificate binds an ID to a public key, managed by an agent called the CA described in following sections. An ACL Entry binds the ID to a set of authorizations, managed by the Security Admin. Authorization management, ACL Entry and Security Admin are as defined in SEMI E132, see Authorizations section.

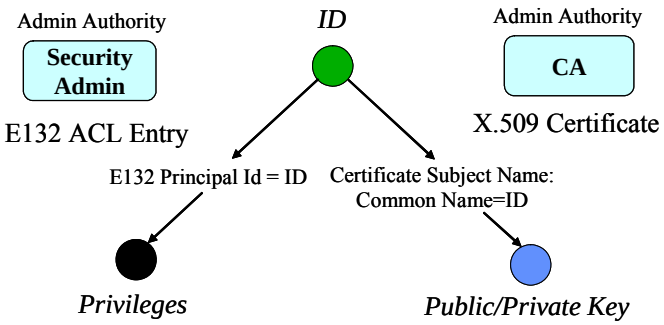

Figure 4
SEMI E132.1 Security Credentials

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

7.2.1.2 CA is the Certification Authority that issues certificates to equipment and application clients wishing to communicate over SSL. The authenticity of the certificate is verified by the CA signature within the certificate, in effect the CA is certifying the validity of the information contained in the certificate, more importantly it certifies the binding between the entity identified in the certificate and its public key used for authentication purposes in SSL. For more information on certificate validation and signature verification, please see referenced X.509 v3 documentation in § 4.6. The CA may or may not be the same as the Security Admin as defined in SEMI E132, which manages access control permissions for application clients. Implementation and maintenance of both the CA and the Security Admin are the responsibility of the equipment user, including certificate issuance and security credentials management. Equipment suppliers shall not be required to provide CA or Security Admin implementations.

7.2.1.3 In this specification, only the common name (CN) attribute of the certificate's subject distinguished name (DN) is required, this is represented as ID in Figure 4. The same ID must be used by the Security Admin when binding authorizations to a principal id in an ACLEntry. In summary, SEMI E132 principals are identified by the common name (CN) attribute of the subject distinguished name field of the certificate issued to the principal.

7.2.1.4 Certificates issued by the CA for SEMI E132 shall be encoded in PKCS#12 format along with corresponding private keys. Equipment shall support PKCS#12 to decode certificate and any associated private key issued by the CA. There are no restrictions on the local encoding format of certificates. See PKCS#12 documentation referenced in § 4.6 for complete details. The CA certificate will also be delivered in same format but without the private key. The CA certificate is used by SSL for certificate verification.

7.2.1.5 Certificates must minimally support signing capability if the KeyUsage extension field is present in the certificate, more specifically the digitalSignature bit must be asserted. Please see the X.509 v3 specification for more information on certificate profile.

7.2.2 *Establishing Equipment Credentials*

7.2.2.1 Equipment must install both the CA's public certificate and its own certificate along with the corresponding private key, the CA certificate to verify client certificates are from the same trusted CA. Equipment is responsible for decoding the PKCS#12 encoded certificates and private key from CA to a format supported by its implementation.

7.2.2.2 An ACLEntry defining the Security Admin for the equipment must be added using the local equipment interface. This shall be done at time of equipment install to allow remote administration of access permissions for the equipment.

7.2.3 *Establishing Application Credentials*

7.2.3.1 Application clients wishing to establish SEMI E132 sessions with any equipment must first possess a valid certificate issued by the CA prior to any communications. Equipment shall ensure SSL handshake terminates for any client that does not present a valid certificate issued by a trusted CA. Similar to equipment certificates, all client certificates and corresponding private key from CA will be encoded in PKCS#12 format. The CA certificate must also be installed by the application client to verify equipment certificate chain.

7.2.3.2 Each client must also have an ACLEntry defined on the equipment granting the appropriate privileges to support its functions prior to establishing SEMI E132 sessions with the equipment. This can be done via the Security Admin or via the equipment console.

7.2.3.3 Equipment shall reject all SEMI E132 establish session requests if client does not have an ACLEntry defined, even if client possesses a valid certificate issued by trusted CA.

7.3 *Enabling/Disabling SSL Authentication*

7.3.1 There may be some integration, development, and/or test environments in which it is not practical or feasible to install equipment and/or client certificates in order to communicate with the equipment. As a practical consideration for such environments only, the equipment shall support the ability of the user to enable or disable SSL authentication at the equipment.

7.3.2 If enabled, all SEMI E132-based communication with the equipment shall take place via mutually-authenticated SSL as described in ¶ 7.1.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310

7.3.3 If disabled, all SEMI E132-based communication with the equipment shall take place via the normal, unauthenticated HTTP transport. While disabled, requests shall still follow the normal authorization verifications using the client ID provided in the FROM field of the SEMI E132 header; however, because requests are not authenticated there can be no guarantee that the requesting application is authentic. Users should be aware that disabling SSL authentication provides unrestricted access to all SEMI E132 equipment operations for any application (factory authorized or not) with network access to the equipment.

7.3.4 Equipment shall provide a configuration option to allow switching between SSL and non-SSL communications. Only one such communication port shall be active at any time. During a configuration change, it is the responsibility of the equipment user to close existing sessions prior to the change, as the active port will no longer be available for requests following the change, and clients will not be notified of the change by the equipment.

7.3.5 After a configuration change, any pre-existing persistent sessions shall remain active as defined in the SEMI E132 Session State Model. Users should be aware that all requests made by those sessions after the change must take place on the new port. The equipment is not required to retain non-persistent sessions in the active state following a configuration change. If an equipment reset is required to change this configuration, then persistent sessions shall be resumed, and non-persistent sessions shall be closed as for any equipment shutdown as defined in the SEMI E132 Session State Model.

8 Mapping of SEMI E132 UML to XML Schema and WSDL

8.1 *WSDL Organization*

8.1.1 Each interface definition in SEMI E132 is mapped to a WSDL portType and binding definition. WSDL portType definitions are named after the interface and its operations as they appear in SEMI E132. WSDL binding definitions for each portType are used to specify the SOAP 1.2 envelope contents for each operation, and to define the corresponding XML encoding styles and HTTP header usage. All SEMI E132 WSDL interfaces use document/literal encoding, with the complete SOAP header and body contents defined in XML Schema file(s) via global element definitions.

8.1.2 Figure 5 shows the relationship between the WSDL binding and portType definitions and the XML Schema types used for each interface. The portType definition imports XML Schema type definitions used in each operation via the XML Schema "import" statement. The WSDL binding definition imports the portType definition via the WSDL "import" statement.

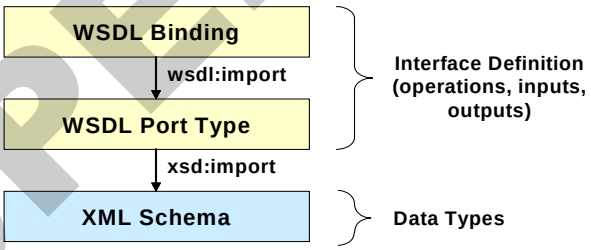

Figure 5
XML Schema and WSDL File Organization

8.1.3 Figure 6 shows the specific XML Schema and WSDL files defined for SEMI E132.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

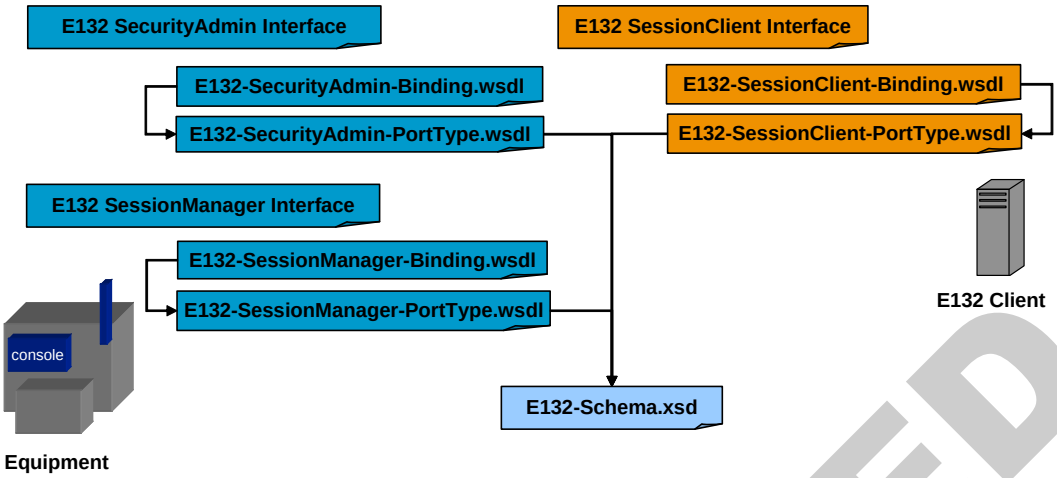

Figure 6
XML Schema and WSDL files for SEMI E132

8.2 *SecurityAdmin Interface*

8.2.1 *Establishing an ACLEntry for the Security Admin*

8.2.1.1 An ACLEntry defining the Security Admin shall be added via the local equipment interface at time of equipment install. The subject id of the ACLEntry must match the common name (CN) attribute of the subject distinguished name of the certificate issued to the Security Admin, see ¶ 7.2.1.3. The Security Admin ACLEntry must grant the reserved privilege "urn:semi-org.auth:securityAdminPrivileges" as defined by SEMI E132.

8.2.1.2 Exactly one ACLEntry shall exist for the Security Admin.

8.2.2 *XML Schema and WSDL Files*

8.2.2.1 The XML Schema and WSDL defined by this specification for the SecurityAdmin interface is contained in the following documents:

Table 9 XML Schema

File Name	E132-1-V0310-Schema.xsd
Target Namespace	urn:semi-org:xsd.E132-1.V0310.auth
Imported/Referenced Namespaces	http://www.w3.org/2001/XMLSchema urn:semi-org:xsd.E138.V0709.ccs urn:semi-org:xsd.E128.V0706.xms
Description	This file defines all of the SEMI E132 data types and elements used by the SecurityAdmin, SessionManager and SessionClient interfaces.

Table 10 SecurityAdmin PortType Definitions

File Name	E132-1-V0310-SecurityAdminPortType.wsdl
Target Namespace	urn:semi-org:ws.E132-1.V0310.SecurityAdminPortType
Imported/Referenced Namespaces	urn:semi-org:xsd.E132-1.V0310.auth http://schemas.xmlsoap.org/wsdl/ http://www.w3.org/2001/XMLSchema urn:semi-org:xsd.E128.V0706.xms
Description	This file defines all of the input/output messages and operations for the SecurityAdmin interface, based on the data types defined in the SEMI E132 XML Schema.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310

Table 11 SecurityAdmin Binding Definitions

File Name	E132-1-V0310-SecurityAdminBinding.wsdl
Target Namespace	urn:semi-org:ws.E132-1.V0310.SecurityAdminBinding
Imported/Referenced Namespaces	urn:semi-org:ws.E132-1.V0310.SecurityAdminPortType http://schemas.xmlsoap.org/wsdl/soap12/ http://schemas.xmlsoap.org/wsdl/
Description	This file binds the abstract portType definition to HTTP and SOAP, specifying required SOAP and HTTP headers for each operation and the XML encoding style for the SecurityAdmin interface.

8.2.2.2 These documents are a part of this specification and should accompany this document. The contents of these documents constitute the core part of this specification.

8.2.3 *WSDL Port Type Overview*

8.2.3.1 The SecurityAdmin WSDL portType definition organizes the SEMI E132.1 XML Schema data types into a collection of named operations with inputs and outputs that correspond to the UML operations that are defined for the SecurityAdmin interface in SEMI E132. The WSDL portType itself is named after the SEMI E132 interface, and each WSDL portType operation is named after the corresponding UML operation defined for the interface. Each WSDL operation consists of two WSDL message definitions corresponding to the input and output (request and response) for the UML operation defined in SEMI E132. Figure 7 illustrates the relationship between the SEMI E132 UML interface and the WSDL portType definition; Table 12 describes this in tabular form. The WSDL message and operation definitions are described in ¶¶ 8.2.9–8.2.15.

8.2.3.2 Detail for the port type operations for SecurityAdmin are defined by the wsdl file in Table 10.

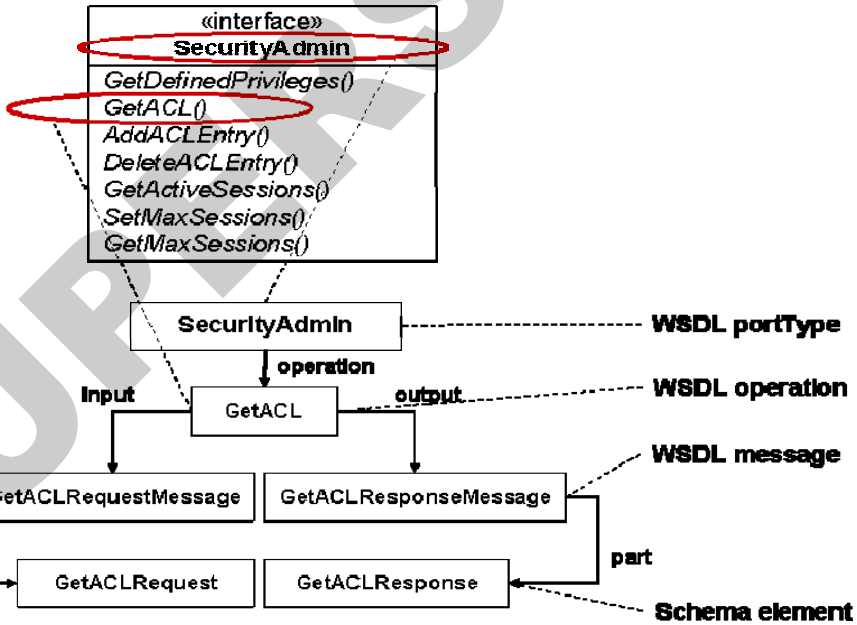

Figure 7
Mapping the SecurityAdmin Interface to a WSDL portType

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E132.1-0310

Table 12 SecurityAdmin Port Type

SEMI E132 Class Name	SecurityAdmin
WSDL Port Type Name	SecurityAdmin
SEMI E132 Operation → WSDL Operation	GetDefinedPrivileges → GetDefinedPrivileges GetACL → GetACL AddACLEntry → AddACLEntry DeleteACLEntry → DeleteACLEntry GetActiveSessions → GetActiveSessions SetMaxSessions → SetMaxSessions GetMaxSessions → GetMaxSessions

8.2.4 *WSDL Binding Overview*

8.2.4.1 The WSDL SecurityAdmin binding definition specifies a message and transport protocol to use for a given portType (SOAP and HTTP for SEMI E132.1), the interface style used (document style for SEMI E132.1). These settings are shown in Table 13. For each WSDL portType operation, the binding defines any SOAP headers used and whether or not they are required, the Content-Type header "action" value to use for that operation, and the XML encoding to use (literal for SEMI E132.1). The binding definitions for each portType operation are described in ¶¶ 8.2.9–8.2.15.

8.2.4.2 Details for the Binding operation services for SecurityAdmin are defined by the file in Table 11. The SOAP action used in all SecurityAdmin operations uses the binding target namespace followed by a colon and the operation name (i.e., soapAction = <binding target namespace>:<operation-name>).

soapAction = "urn:semi-org:ws.E132-1.V0310.SecurityAdminBinding:<operation-name>"

Table 13 SecurityAdmin Binding

SEMI E132 Class Name	SecurityAdmin
WSDL Binding Name	SecurityAdminBinding
SOAP Binding Style	document
SOAP Transport	http://schemas.xmlsoap.org/soap/http

8.2.5 *WSDL Service Overview*

8.2.5.1 This specification does not provide a WSDL service definition. WSDL service definitions provide one way to define and locate the endpoint(s) at which a given interface can be accessed by a client. Such information cannot be determined until the equipment has been installed on a factory network, and therefore is out of scope for this specification. Users are responsible for defining the mechanisms by which clients locate SEMI E132 equipment services in the factory. A sample SecurityAdmin service definition is provided in Figure 8 for reference only.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E132.1-0310