Objectives - 1

- To create an XML Schema Definition (XSD) specification that enables the electronic notarization, witnessing and apostillization of documents – both XML-based and non-XML based
- To enable many signature technologies to be used with the specification
Objectives - 2

- Create XML e-Notarization element(s) that can be embedded into existing XML documents to depict a notarized document.
- Create a standard XML container for e-Notarized documents that can contain any document – including those created by legacy (XML-unaware) applications.
Objectives - 3

- XML documents using the eNotary XSD specification must be capable of being used with the:
  - Null Signature Profile
  - Symmetric Key Signature Profile
  - OASIS Digital Signature Services (DSS) protocol
  - W3C XML Signature standard
Objectives - 4

- **If possible**, address the need to accommodate Witnessed Documents and Apostilles
  - Witnessed documents are defined to be documents signed by one or more Signers, and witnessed and signed by a notary public, but without a notarial certificate

- **If possible**, allow the XSD to be usable by international notaries
Solution

- The OASIS LegalXML eNotarization Technical Committee's eNotary Specification 1.0 (DRAFT 7), meets **ALL** these objectives

....and more!
e-Notary element for XML documents

Legend:

- **SomeElement**: A simple element that does NOT contain any sub-elements
- **SomeElement**: A complex element that contains sub-elements
- **ChoiceElement**: A choice between one or more elements
Objective

- To create XML e-Notarization element(s) which can be embedded into existing XML documents to depict a notarized document
- The enclosing XML document schema should not have to change, except for the addition of the optional eNotary elements
Before adding e-Notary elements - 1
<example:SomeProprietaryXMLDocument
    xmlns:xenc="http://www.example.org/2001/example#" >
  <example:SomeElement1>...</example:SomeElement1>
  <example:SomeElement2>
    <example:SomeElement3>...</example:SomeElement3>
    <example:SomeElement4>...</example:SomeElement4>
    <example:SomeElement5>...</example:SomeElement5>
    <example:SomeElement6>...</example:SomeElement6>
  </example:SomeElement2>
  <example:SomeElement7>...</example:SomeElement7>
  ...
  ...
  <example:SomeElement17>...</example:SomeElement17>
</example:SomeProprietaryXMLDocument>
After adding e-Notary elements - 1

<table>
<thead>
<tr>
<th>Some Proprietary XML Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some Element 1</td>
</tr>
<tr>
<td>Some Element 2</td>
</tr>
<tr>
<td>Some Element 3</td>
</tr>
<tr>
<td>Some Element 4</td>
</tr>
<tr>
<td>Some Element 5</td>
</tr>
<tr>
<td>Some Element 6</td>
</tr>
<tr>
<td>Some Element 7</td>
</tr>
<tr>
<td>Some Element 8</td>
</tr>
<tr>
<td>Some Element 9</td>
</tr>
<tr>
<td>Some Element 10</td>
</tr>
<tr>
<td>Some Element 11</td>
</tr>
<tr>
<td>Some Element 12</td>
</tr>
<tr>
<td>Some Element 13</td>
</tr>
<tr>
<td>Some Element 14</td>
</tr>
<tr>
<td>Some Element 15</td>
</tr>
<tr>
<td>Some Element 16</td>
</tr>
<tr>
<td>Some Element 17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notary Certificates</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Notary Signatures</th>
</tr>
</thead>
</table>

Change #1: Instead of just a Notary Certificates element, proprietary XML must also include a Notary Signatures element.
After adding e-Notary elements - 2

<example:SomeProprietaryXMLDocument
    xmlns:xenc="http://www.example.org/2001/example#"
    <example:SomeElement1>...</example:SomeElement1>
    <example:SomeElement2>
        <example:SomeElement3>...</example:SomeElement3>
        <example:SomeElement4>...</example:SomeElement4>
        <example:SomeElement5>...</example:SomeElement5>
        <example:SomeElement6>...</example:SomeElement6>
    </example:SomeElement2>
    <example:SomeElement7>...</example:SomeElement7>
    ...
    <example:SomeElement17>...</example:SomeElement17>
    <enml:NotaryCertificates>...</enml:NotaryCertificates>
    <enml:NotarySignatures>...</enml:NotarySignatures>
</example:SomeProprietaryXMLDocument>
The Notary Certificates element contains a minimum of one, but can contain any number of individual Notary Certificate elements.

Note: This addresses the corner case where a single document may be notarized by many Notaries Public for multiple signers. Each Notary Certificate is tied to a specific set of elements and can be notarized independently.
Contents of Notary Signatures

The Notary Signatures element contains a minimum of one, but can contain any number of individual Signature or Notary Signature elements.

Note: Applications must ensure that there is a unique sub-element inside Notary Signatures for each Notary Certificate that exists in the Notary Certificates element.
Each Notary Certificate element contains two (2) sub-elements:

1) The Certificate Content element that has required statutory information about the notarization.

Change #2: There is now a choice of specifying if the notarization is being done at an international or US location.
2) The Notary Public container element contains required information about the Notary Public.

Change #3: There is now a choice of specifying if the jurisdiction in which the Notary can perform their duties is an international or US location.

Change #4: There is now an optional Notary Bond Number element.

Change #5: There is an optional element to specify a URI that allows an application to verify information about the Notary Public through a web-service.
The Notary Signatures element contains a choice of either a W3C XML Signature element or an element of the “Any” type that can carry a Notary Signature element.

The XML Signature element can accommodate the Symmetric Key profile, the DSS profile and the XML Signature profile.

The Notary Signature element will accommodate any new signature type element that isn't supported today.
### Some Proprietary XML Document

<table>
<thead>
<tr>
<th>Some Element 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some Element 2</td>
</tr>
<tr>
<td>Some Element 3</td>
</tr>
<tr>
<td>Some Element 4</td>
</tr>
<tr>
<td>Some Element 5</td>
</tr>
<tr>
<td>Some Element 6</td>
</tr>
<tr>
<td>Some Element 7</td>
</tr>
<tr>
<td>Some Element 8</td>
</tr>
<tr>
<td>Some Element 9</td>
</tr>
<tr>
<td>Some Element 10</td>
</tr>
<tr>
<td>Some Element 11</td>
</tr>
<tr>
<td>Some Element 12</td>
</tr>
<tr>
<td>Some Element 13</td>
</tr>
<tr>
<td>Some Element 14</td>
</tr>
<tr>
<td>Some Element 15</td>
</tr>
<tr>
<td>Some Element 16</td>
</tr>
<tr>
<td>Some Element 17</td>
</tr>
</tbody>
</table>

### Notary Certificates

#### Notary Certificate

- Certificate Content
  - NotarizationType
  - NotarizationDate
  - NotarizationInternationalLocation
  - NotarizationUSLocation
  - StatutoryText

#### Notary Public

- Notary Name
- Notary Commission Number
- Notary Commission Expiry Date
- Notary International Jurisdiction
- Notary US Jurisdiction
- Notary Bond Number
- Notary Verification URI

#### Notary Signatures

- Signature
- Notary Signature
e-Notary Element Prerequisites

- Application must already know how to read, process and store XML content
- XML content created by application must have the other prerequisite elements required for e-Notarization
  - The document being notarized
  - The document Signers' electronic signatures
e-Notary Element Features

- Useful for applications that have their own XML schema (or choose to use their own proprietary schema) and need to just add e-Notarization capability to it (+)
- XML document with e-Notary elements can only be interpreted by applications that understand the application's proprietary XML schema (-)
Example of how it would work

Legend:

SomeElement: A simple element that does NOT contain any sub-elements
SomeElement: A complex element that contains sub-elements
ChoiceElement: A choice between one or more elements
We begin with a document based on a proprietary XML schema that has **one** Signer.

<table>
<thead>
<tr>
<th>Some Proprietary XML Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some Element 1</td>
</tr>
<tr>
<td><strong>Some Element 2</strong></td>
</tr>
<tr>
<td>Some Element 3</td>
</tr>
<tr>
<td>Some Element 4</td>
</tr>
<tr>
<td>Some Element 5</td>
</tr>
<tr>
<td>Some Element 6</td>
</tr>
<tr>
<td>Some Element 7</td>
</tr>
<tr>
<td>Some Element 8</td>
</tr>
<tr>
<td>Some Element 9</td>
</tr>
<tr>
<td>Some Element 10</td>
</tr>
<tr>
<td>Some Element 11</td>
</tr>
<tr>
<td>Some Element 12</td>
</tr>
<tr>
<td>Some Element 13</td>
</tr>
<tr>
<td><strong>Signer1 at Location 1</strong></td>
</tr>
</tbody>
</table>
The first Notary Certificate & Signature is added to the document, acknowledging the first signature.

This notarial certificate notarizes the XML document and the first Signer.

The signature in this element covers this document, and the first Signer element.
The document is sent to Location 2 where two new Signers sign the document. These elements are added to the document, as shown below.
The second Notary Certificate now covers the document and the two new Signers' signatures. This can continue ad infinitum.

The first certificate with the first signature continues to validate the document and the first Signer.

The second certificate with the second signature now validates the document and the second/third Signers.
Important note about subsequent Notary Certificates in documents

- Applications must not fall into the trap of assuming that a second (third or fourth..) Notary Certificate automatically notarizes earlier Signers and their signatures in the document.

- Applications **MUST** validate each document signers' signature **ONLY** with the appropriate referenced Notary Certificate & Signature.
e-Notarized Document

Legend:

*SomeElement*: A simple element that does NOT contain any sub-elements

*SomeElement*: A complex element that contains sub-elements

*ChoiceElement*: A choice between one or more elements
Where e-Notary element does not help

- If the document was created using an application that does not know XML
- If the document is a scanned image of a paper document (GIF, JPG, etc.)
- If the document is being e-notarized after it was created and cannot be processed through an e-Notary element-aware application
Where e-Notary element does not help

- If an application does not want to use (or does not understand) someone else's proprietary XML schema and needs an industry-standard container for e-notarized documents
Solution

- An e-Notary container with the original electronic document embedded in it
- Original document is Base-64 encoded inside the e-notarized document, without modifying its content
- The standard e-Notary elements are used to notarize the document
Features

- Works for all legacy documents and applications (that do not know how to create XML or how to add the e-Notary element to existing XML) (+)

- Requires a separate tool to create the new notarized XML document (-)*

* It is anticipated that these tools will show up once the standard is created.
The Notarized Document. Is the outermost element container for the legacy e-Notarized documents.

It contains four (4) sub-elements.
The first is a container element for Signed Documents which contains a Signed Document.
Notarized Document

Signed Documents

Document Signers

The second is a container element for document's signers.
Notarized Document

Signed Documents

Notary Certificates

Document Signers

The third is the container element for Notary Certificates.
Notarized Document

- Signed Documents
- Notary Certificates
- Document Signers
- Notary Signatures

The fourth is the container element for Notary Signatures.
The **Signed Documents** container contains a **Signed Document** element, which contains four (4) sub-elements. They are:
1. The Base64-encoded document being notarized.

Note: The encoding does NOT change the document and allows ANY type of document to be contained in this element.
2. An element to identify the MIME type of the document, so applications know how to deal with it.
<table>
<thead>
<tr>
<th>Signed Documents</th>
<th>Notary Certificates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signed Document</strong></td>
<td>Document Signers</td>
</tr>
<tr>
<td><em>Document</em></td>
<td>DocumentMIMETYPE</td>
</tr>
</tbody>
</table>

3. An **optional** comments element for a hint about the MIME-type of the document, if it isn't recognized as a known MIME-type.
4. An optional comments element for any notes about the document itself.
The Document Signers element contains, at least, one Document Signer element that provides details about the document-signer. There may be as many Document Signer elements in this container as needed.
Change #6: There is now an element to record the Signer’s title.

Change #7: There is now a choice of specifying the location of the signee, if any is required. It can either be an international or US location.

Change #8: There is now a Signer Identification Method element, as opposed to a Signer ID element. No private information is recorded now in a eNotary document.

Change #9: The signature of the Signer is now a choice of an XML Signature or an element of “Any” type.
The Notary Certificates element is the same element embedded inside “proprietary” XML documents.

The Notary Signatures element is the same element embedded inside “proprietary” XML documents.
Notarized Document

Signed Documents

Signed Document

Document
DocumentMIMETYPE
DocumentMIMETYPEComments
DocumentComments

Notary Certificates

Notary Certificate

Certificate Content
NotarizationType
NotarizationDate
NotarizationInternationalLocation
NotarizationUSLocation
StatutoryText

Notary Public

Notary Name
Notary Commission Number
Notary Commission Expiry Date
Notary International Jurisdiction
Notary US Jurisdiction
Notary Bond Number
Notary Verification URI

Document Signers

Document Signer

Signer Name
Signer Title
Signer International Address
Signer US Address
Signer Identification Method

Signer Signature

Notary Signatures

Signature
Notary Signature
Example of how it would work

Legend:

*SomeElement*: A simple element that does NOT contain any sub-elements

*SomeElement*: A complex element that contains sub-elements

*ChoiceElement*: A choice between one or more elements
We begin with a document inside the OASIS Signed Document container with one Document Signer.
The document is notarized by encapsulating it into an OASIS Notarized Document container, with a Notary Certificates and Notary Signatures elements.

Notarized Document
Signed Documents
Signed Document
- Document
- DocumentMIMEType
- DocumentMIMETypeComments
- DocumentComments

Document Signers
Document Signer (1)

Notary Certificates
Notary Certificate (1)
- Certificate Content
- Notary Public

Notary Signatures
Signature (1)

The signature in this element covers the encapsulated document, the signer element, and the first notarial certificate.
The document is sent to another location where the second signers' signature is added to the document.
The document is notarized by another Notary, covering the original document and the second Signers' signature.

The signature in the first Notary Public element continues to validate the document and the first Signer.

The signature in the second Notary Public element covers the same document, the second notarial certificate and the second signer element.
Summary

- The e-Notary elements and the Notarized Document container, allow all types of documents to take advantage of the standards:
  - Those that can embed the e-Notary element into XML; and
  - Those that cannot (legacy documents) or choose not to
e-Witnessed Document

Legend:

SomeElement: A simple element that does NOT contain any sub-elements
SomeElement: A complex element that contains sub-elements
ChoiceElement: A choice between one or more elements
Definition of a Witnessed Document

An electronic document, legal in many US states, and:
- Signed by one or more Signers
- Signing is witnessed by a Notary Public
- Document is signed by Notary Public
- Document is recorded in Notary Journal
- BUT there is NO Notarial Certificate
Top-level container is a Witnessed Document as opposed to a Notarized Document.

Signed Documents element and all its contents are identical to the Notarized Document element.

Instead of the Notary Certificates element, we have a Notaries Public element, which has one or more of the same Notary Public elements.

Finally, we have the Notary Signatures element which allows the notaries to witness and sign the document.

However, because there is no Notary Certificate element, this is now a valid Witnessed Document.
What else can this schema do

- While the law currently does not permit notarizing multiple paper documents with a single document signer's signature or with a single Notary's signature, this schema allows for notarizing/witnessing multiple electronic documents with a single electronic signature of the Document Signer and/or the Notary Public.
Benefits of single-selling

- **Increased productivity**
  - Saves time for everyone – a single electronic signature from the Document Signer and from the Notary Public can notarize/witness the entire set of electronic documents within a transaction

- **Increased savings**
  - Saves money for everyone (except, perhaps, for the Notary) since multiple documents can be processed in one pass

A single notarial certificate covers all Signed Document elements and the signer's signature in this notarized document.

A single notary signature covers all Signed Document elements, the signer's signature and the notarial certificate in this notarized document.

Signed Documents container can contain one or unlimited number of Signed Document elements.
Conclusion

- The e-Notary schema accommodates notarized and witnessed documents
- The e-Notary schema accommodates US and international locations
- The e-Notary schema handles Null signatures, symmetric-key signatures and XML Signatures
Conclusion

- The e-Notary schema can handle getting XML Signatures from a Digital Signature Service (DSS) protocol-based service.
- The e-Notary schema can improve the notarization process in the future and make signings/verifications more productive while lowering costs.
Thank you

Please send comments to:

- legalxml-enotary-comment@lists.oasis-open.org
- arshad.noor@strongauth.com