

The following slides explain how two types of Notarized Documents - regardless of whether they are Acknowledgments, Jurats, or other type of Notarized Documents - are represented in XML. A concerted effort was made to keep the design of the elements simple while providing flexibility in implementation for applications using existing XML schemas or applications that do not know how to process XML.

Please send comments to author.

Document Name: eNotary-XSchema-Overview-4

Date Created: January 27, 2008

Date Modified: March 14, 2008

Creator: Arshad Noor

E-mail: arshad.noor@strongauth.com

Legend:

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

SomeElement: A complex element that contains sub-elements

Objectives

- Create an XML e-Notarization element 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

e-Notary element for XML documents

Before adding e-Notary element - 1

Some Proprietary XML Document

Some Element 1

Some Element 2

Some Element 3

Some Element 4

Some Element 5

Some Element 6

Some Element 7

Some Element 8

Some Element 9

Some Element 10

Some Element 11

Some Element 12

Some Element 13

Some Element 14

Some Element 15

Some Element 16

Some Element 17

Before adding e-Notary element - 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>
</example:SomeProprietaryXMLDocument>
```

After adding e-Notary element - 1

Some Proprietary XML Document

Some Element 1

Some Element 2

Some Element 3
Some Element 4
Some Element 5
Some Element 6

Some Element 7

Some Element 8

Some Element 9

Some Element 10

Some Element 11

Some Element 12

Some Element 13

Some Element 14

Some Element 15

Some Element 16

Some Element 17

Notary Certificates

After adding e-Notary element - 2

```
<example:SomeProprietaryXMLDocument
  xmlns:xenc="http://www.example.org/2001/example#"
  xmlns:xenc="http://www.oasis-open.org/2008/enotary#">
  <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>
  <enotary:NotaryCertificates>...</enotary:NotaryCertificates>
</example:SomeProprietaryXMLDocument>
```

Contents of e-Notary element

Some Proprietary XML Document

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.

Notary Certificates

Notary Certificate

Notary Certificate

•

•

•

Some Element 15

Some Element 16

Some Element 17

Some Proprietary XML Document

Each Notary Certificate element contains three (3) sub-elements:

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

Some Element 11

Some Element 12

Some Element 13

Some Element 14

Some Element 15

Some Element 16

Some Element 17

Notary Certificates

Notary Certificate

Certificate Content

NotarizationType
NotarizationDate
NotarizationLocation
StatutoryText

Some Proprietary XML Document

Some Element 1

Some Element 2

Some Element 3

Some Element 4

2) The Notary Public container element contains requisite information about the Notary Public.

Some Element 10

Some Element 11

Some Element 12

Some Element 13

Some Element 14

Some Element 15

Some Element 16

Some Element 17

Notary Certificates

Notary Certificate

Certificate Content

NotarizationType

NotarizationDate

NotarizationLocation

StatutoryText

Notary Public

Notary Name

Notary Commission Number

Notary Commission Expiry Date

Notary Jurisdiction

Notary Digital Certificate

Some Proprietary XML Document

Some Element 1

Some Element 2

Some Element 3

Some Element 4

Some Element 5

3) The Notary Signature element contains the Notary's signature conforming to the XML Signature standard, which references the elements that have been notarized.

Some Element 15

Some Element 16

Some Element 17

Notary Certificates

Notary Certificate

Certificate Content

NotarizationType

NotarizationDate

NotarizationLocation

StatutoryText

Notary Public

Notary Name

Notary Commission Number

Notary Commission Expiry Date

Notary Jurisdiction

Notary Digital Certificate

Notary Signature

The full view

Some Proprietary XML Document

Some Element 1

Some Element 2

Some Element 3
Some Element 4
Some Element 5
Some Element 6

Some Element 7

Some Element 8

Some Element 9

Some Element 10

Some Element 11

Some Element 12

Some Element 13

Some Element 14

Some Element 15

Some Element 16

Some Element 17

Notary Certificates

Notary Certificate

Certificate Content

NotarizationType
NotarizationDate
NotarizationLocation
StatutoryText

Notary Public

Notary Name

Notary Commission Number

Notary Commission Expiry Date

Notary Jurisdiction

Notary Digital Certificate

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
 - Signers' identification information
 - 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 element can only be interpreted by applications that understand the application's proprietary XML schema (-)

Example of how it would work

We begin with a document based on a proprietary XML schema that has one Signer.

Some Proprietary XML Document

Some Element 1

Some Element 2

Some Element 3
Some Element 4
Some Element 5
Some Element 6

Some Element 7

Some Element 8

Some Element 9

Some Element 10

Some Element 11

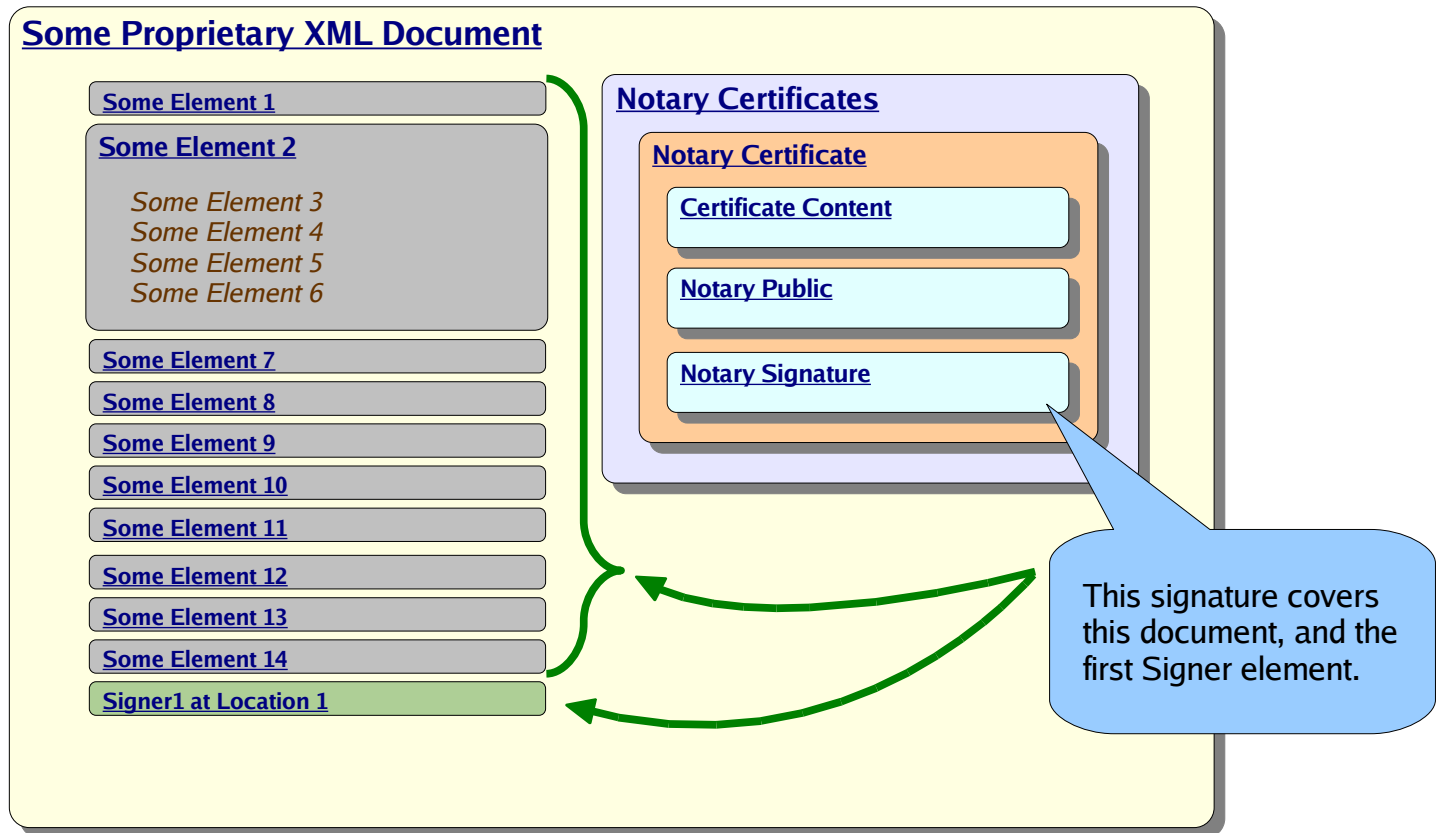
Some Element 12

Some Element 13

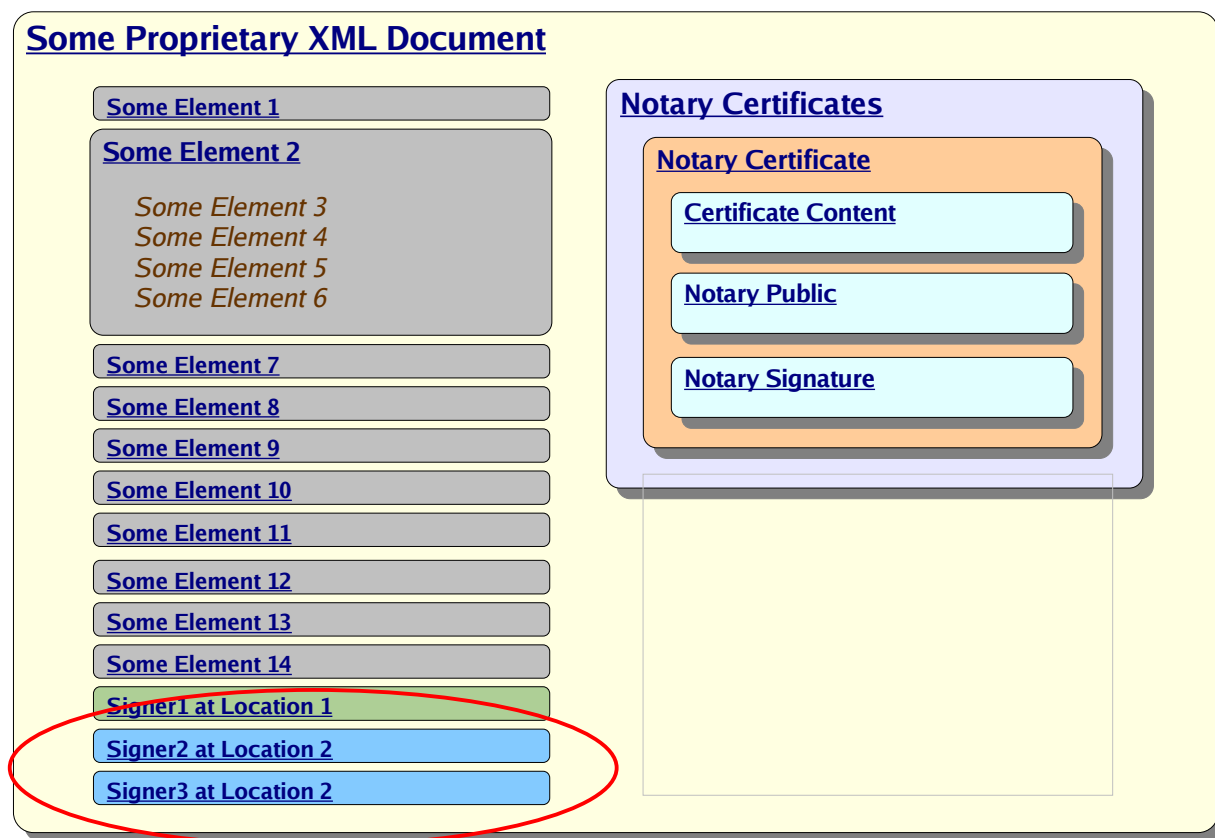
Some Element 14

Signer1 at Location 1

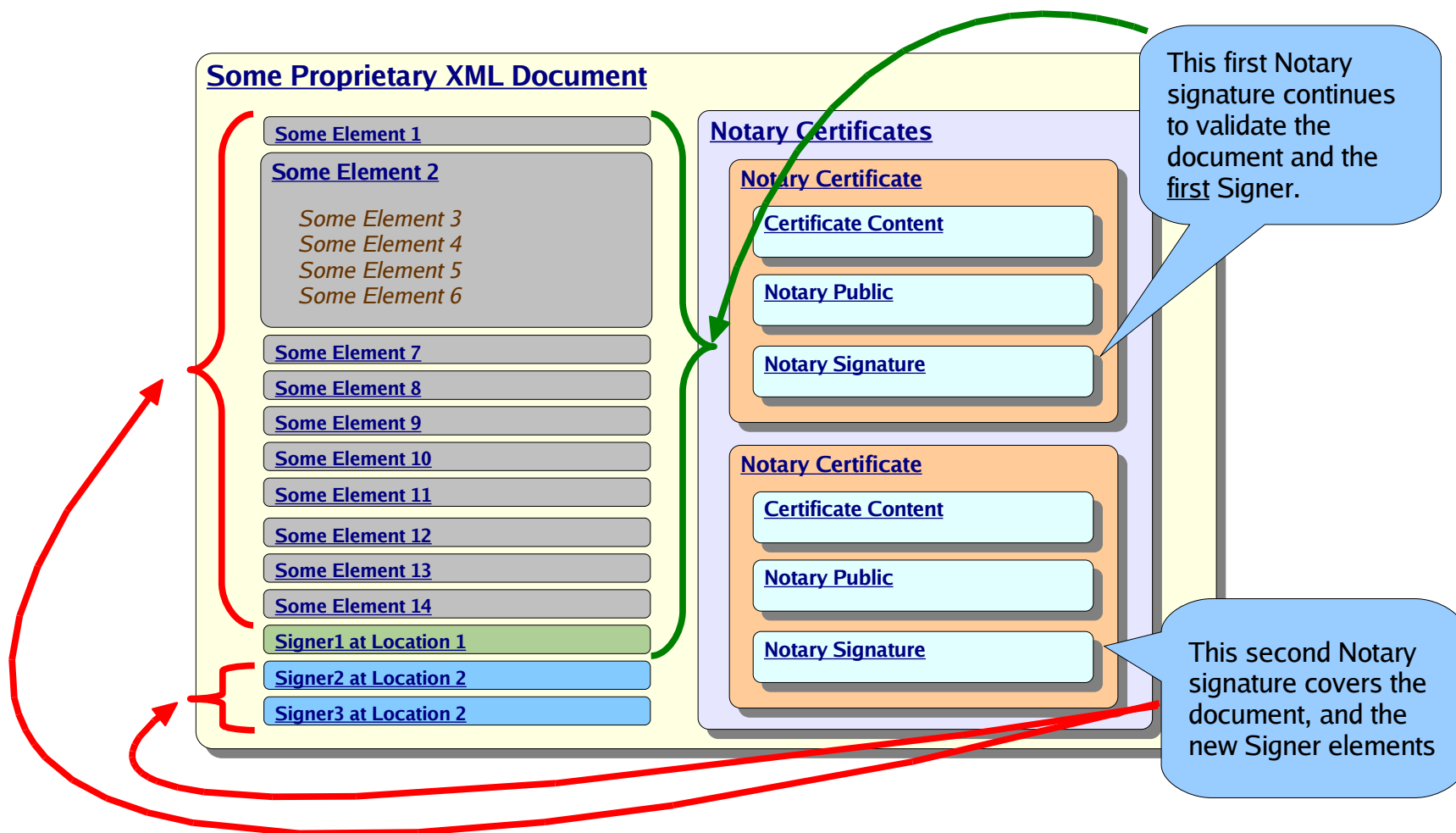
The first Notary Certificate is added to the document, acknowledging the first signature.



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.



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 Signers' signature **ONLY** with the Notary Certificate that references the signature to be deemed valid

e-Notarized Document

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 element is 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.

Notarized Document

The Notarized Document.
Is the outermost element
container for the legacy
e-Notarized documents.

It contains two (2) sub-
elements.

The first is a container element for the Signed Document.

Notarized Document

Signed Document

Notarized Document

Signed Document

Notary Certificates

The second is a container element for the Notary Certificates.

Notarized Document

Signed Document

Notary Certificates

The **Signed Document** element contains four (4) types of elements. They are:

Notarized Document

Signed Document

Document

Notarized

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.

Notarized Document

Signed Document

Document
DocumentMIMEType

Nota

2. An element to identify the MIME type of the document, so applications know how to deal with it.

Notarized Document

Signed Document

Document
DocumentMimeType
DocumentComments

Notary Certificates

3. An **optional** comments element for notes about the document, if any.

Notarized Document

Signed Document

Document
DocumentMIMEType
DocumentComments

Signer

Signer Name

Signer Address

Signer ID

Signer Digital Certificate

Signature

.....

Signer

Nota

4. At least one Signer element (although there may be as many Signer elements as are needed) that provides details about the document-signer(s).

The Signer Digital Certificate and Signature are **optional** elements, as the signature may be embedded in the document itself.

Notarized Document

The Notary Certificates element is the same element embedded inside “proprietary” XML documents.

Signer

.....

Notary Certificates

Notary Certificate

Certificate Content

NotarizationType
NotarizationDate
NotarizationLocation
StatutoryText

Notary Public

Notary Signature

Notarized Document

Signed Document

Document
DocumentMIMEType
DocumentComments

Signer

Signer Name

Signer Address

Signer ID

Signer Digital Certificate

Signature

.....

Signer

Notary Certificates

Notary Certificate

Certificate Content

NotarizationType
NotarizationDate
NotarizationLocation
StatutoryText

Notary Public

Notary Name

Notary Commission Number

Notary Commission Expiry Date

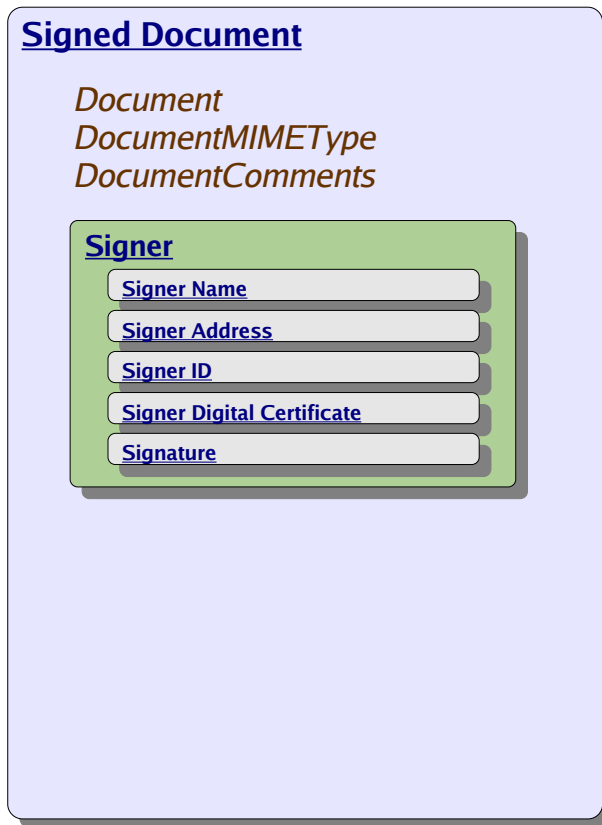
Notary Jurisdiction

Notary Digital Certificate

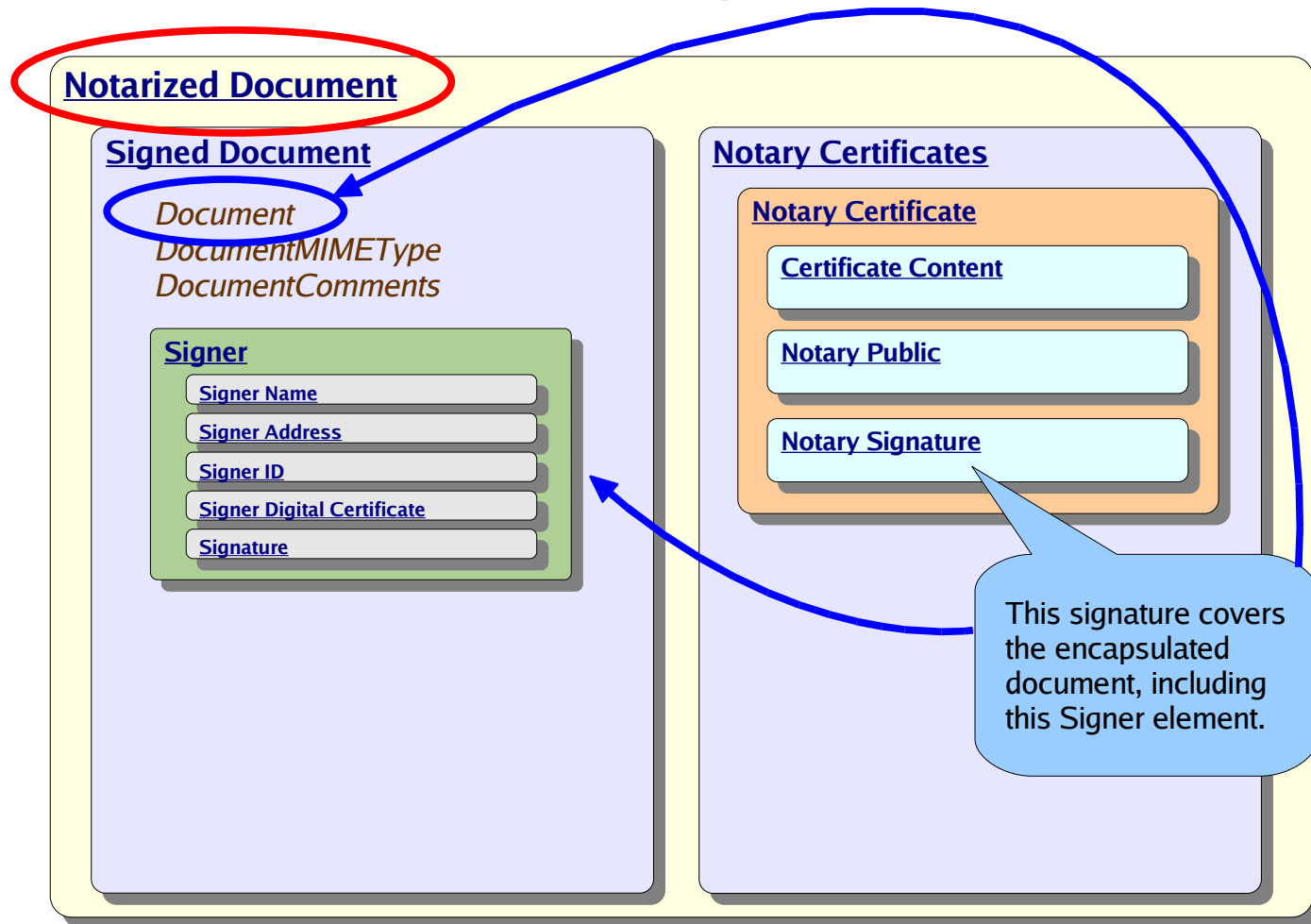
Notary Signature

Example of how it would work

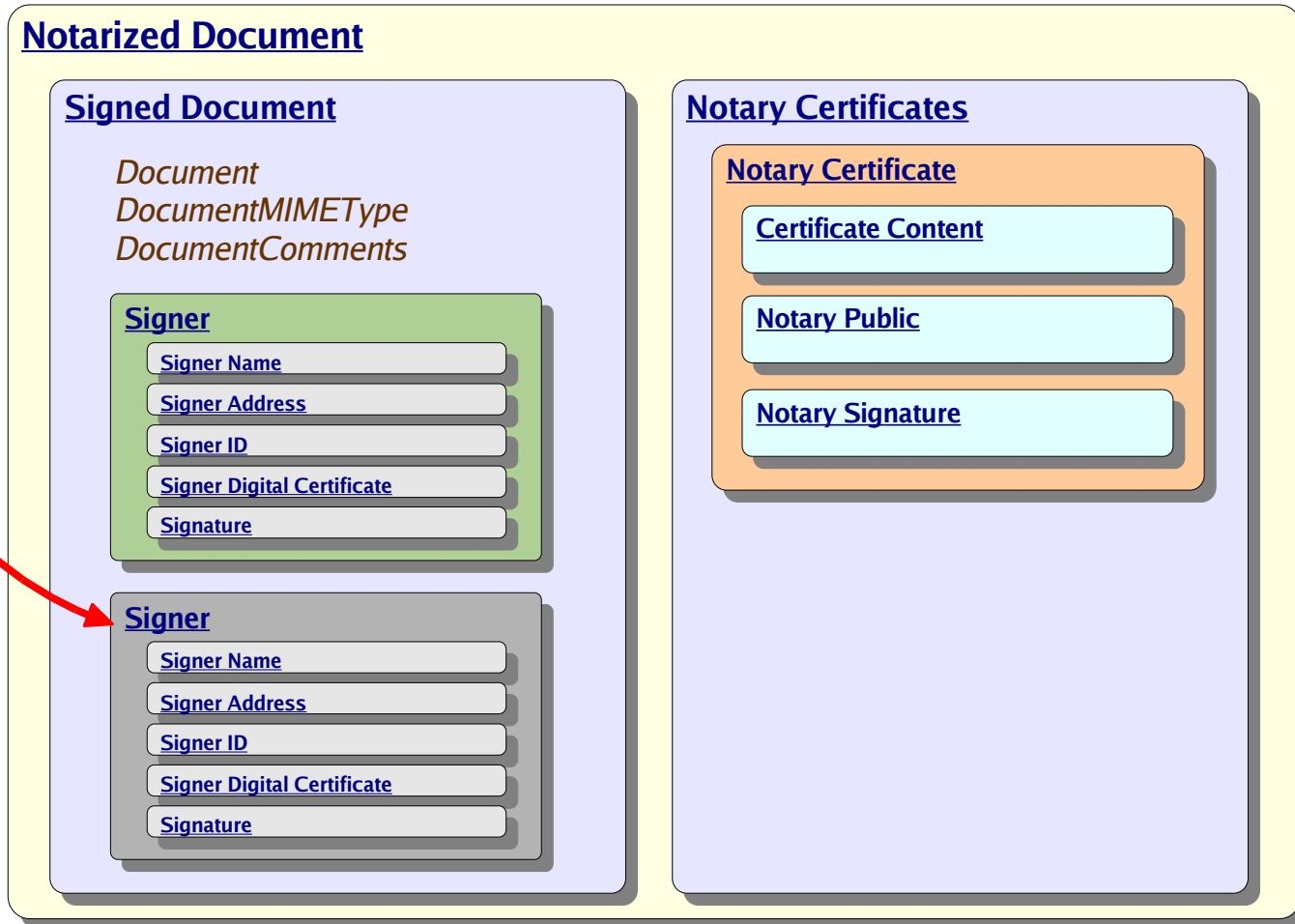
We begin with a document encapsulated into an OASIS Signed Document container with one Signer.



The document is notarized by encapsulating it in an OASIS Notarized Document container with a Notary Certificates element containing one Notary Certificate.



The document is to another location where the second Signers' signature is added to the document.



Notarized Document

Signed Document

Document
DocumentMimeType
DocumentComments

Signer

Signer Name
Signer Address
Signer ID
Signer Digital Certificate
Signature

Signer

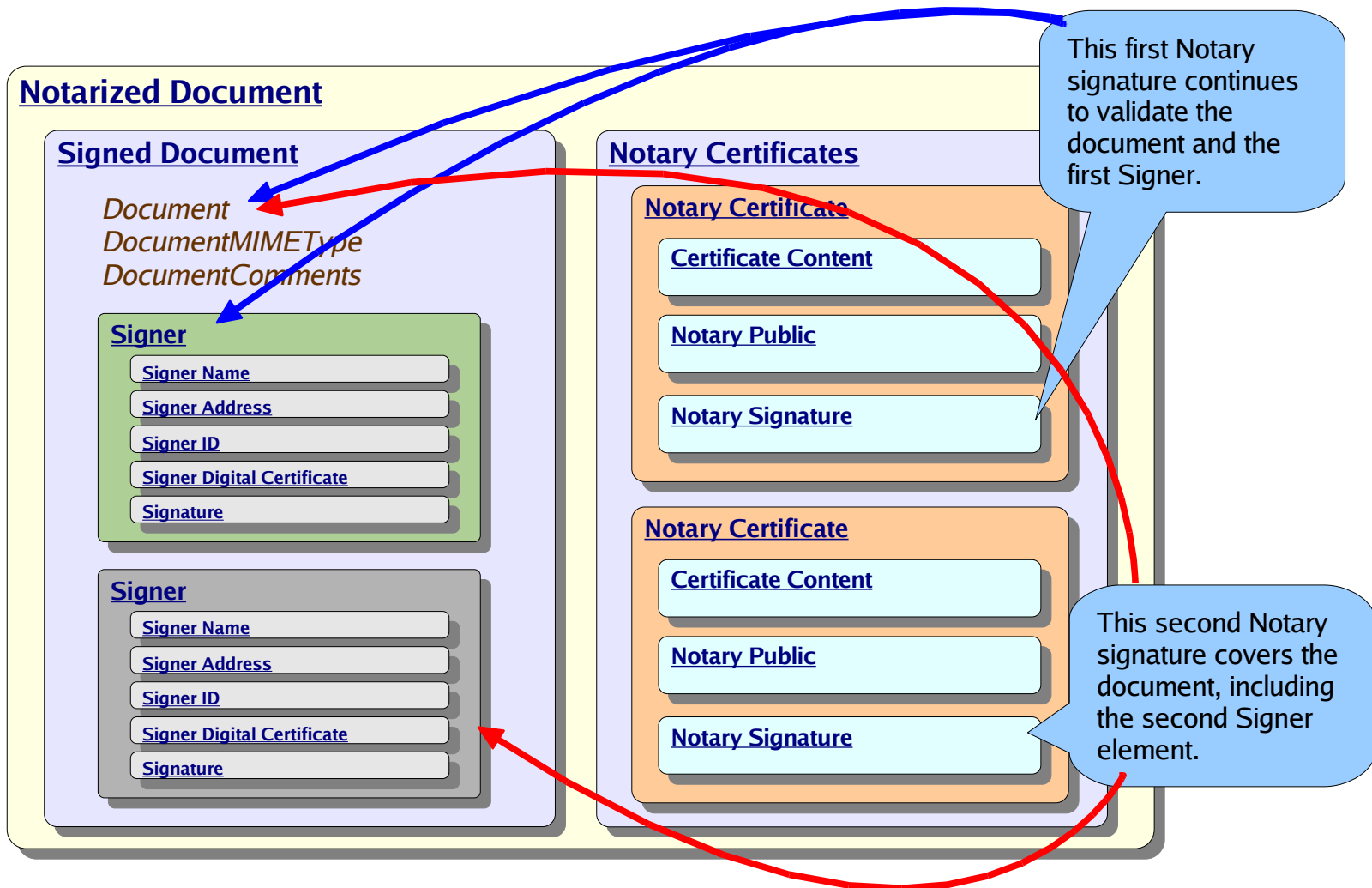
Signer Name
Signer Address
Signer ID
Signer Digital Certificate
Signature

Notary Certificates

Notary Certificate

Certificate Content
Notary Public
Notary Signature

The document is notarized by another Notary, covering the original document and the second Signers' signature.



Conclusion

- The two standards, the e-Notary element and the e-Notarized Document, 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