

The IHE logo consists of the letters 'IHE' in a bold, dark blue, sans-serif font. A vertical line is positioned to the right of the letters, separating them from the tagline.

Integrating
the Healthcare
Enterprise

New Asynchronous AS4 Web Services Option for XCA, XCPD, XDR and XCDR Profiles

Presented by Charles Parisot
GE Healthcare, IHE ITI Technical Committee Member,
IHE Intl Board Member

Interest of Asynchronous AS4 Web Services

To support scaling of document sharing between communities to a large numbers of gateways, Asynchronous Web Services Exchange is critical to realize a more efficient handling of latency and scale.

Asynchronous Web Services with AS4 relies on the OASIS AS4 WS Stack that has been natively designed to support Asynchronous WS exchange and offers:

- Message packaging governed by ebMS 3.0 and message security governed by WS-Security
- Support for both *push* and *pull* message exchange choreographies
- Optional Payload compression
- Non-Repudiation of Origin and Receipt
- Reception Awareness – simple and effective reliable messaging with no known interoperability issues

Approach to the Asynchronous AS4 Web Services Option

A new Supplement issued by the IHE IT Infrastructure Committee in August 2018, which introduces:

- a. An option to the profiles where the current WS-Addressing based Asynchronous WS Exchange is currently available:
 - XCA: Cross Community Access for sharing documents
 - XCPD Cross Community Patient Discovery
- b. A new option to the XDR and XCDR Profiles, where the ITI-41 is specified to support Asynchronous WS, but the option is not explicitly stated as an Option.
- c. An alternative to the current WS-Addressing based Asynchronous Option

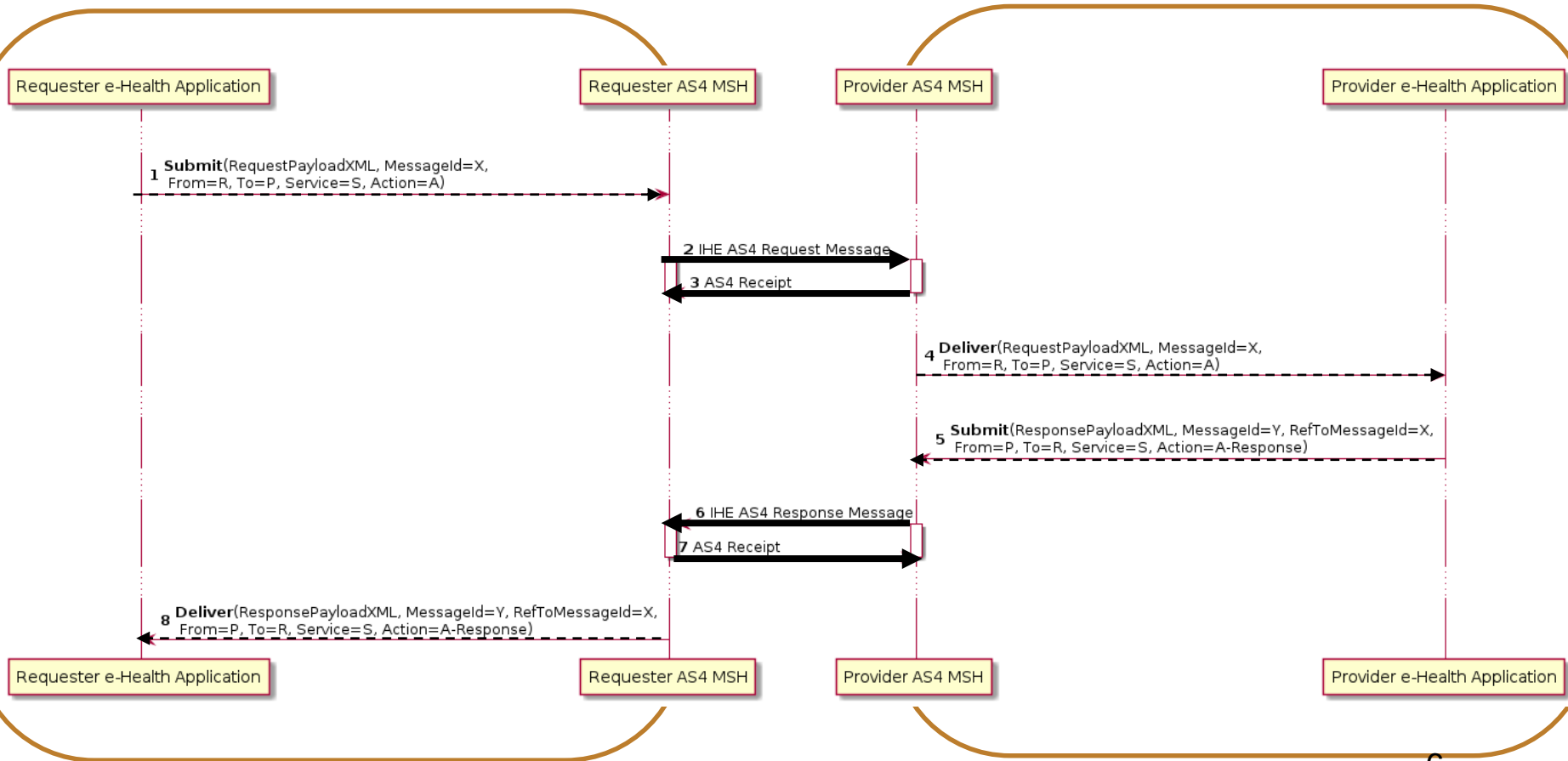
Main Features of Asynchronous AS4

- Based on pre-defined AS4 Conformance Profiles:
 - ebHandler (messaging client and server), or
 - Light Client (messaging client only)
- Security
 - Signing and encryption using WS-Security, XML Signature, XML Encryption
 - IHE leaves details (algorithms etc.) to projects
- Reliable Messaging
 - AS4 reception awareness (receipts, retries, duplicate elimination)
- Error Handling and Receipt Handling
 - For push, profiled to use synchronous errors/receipts

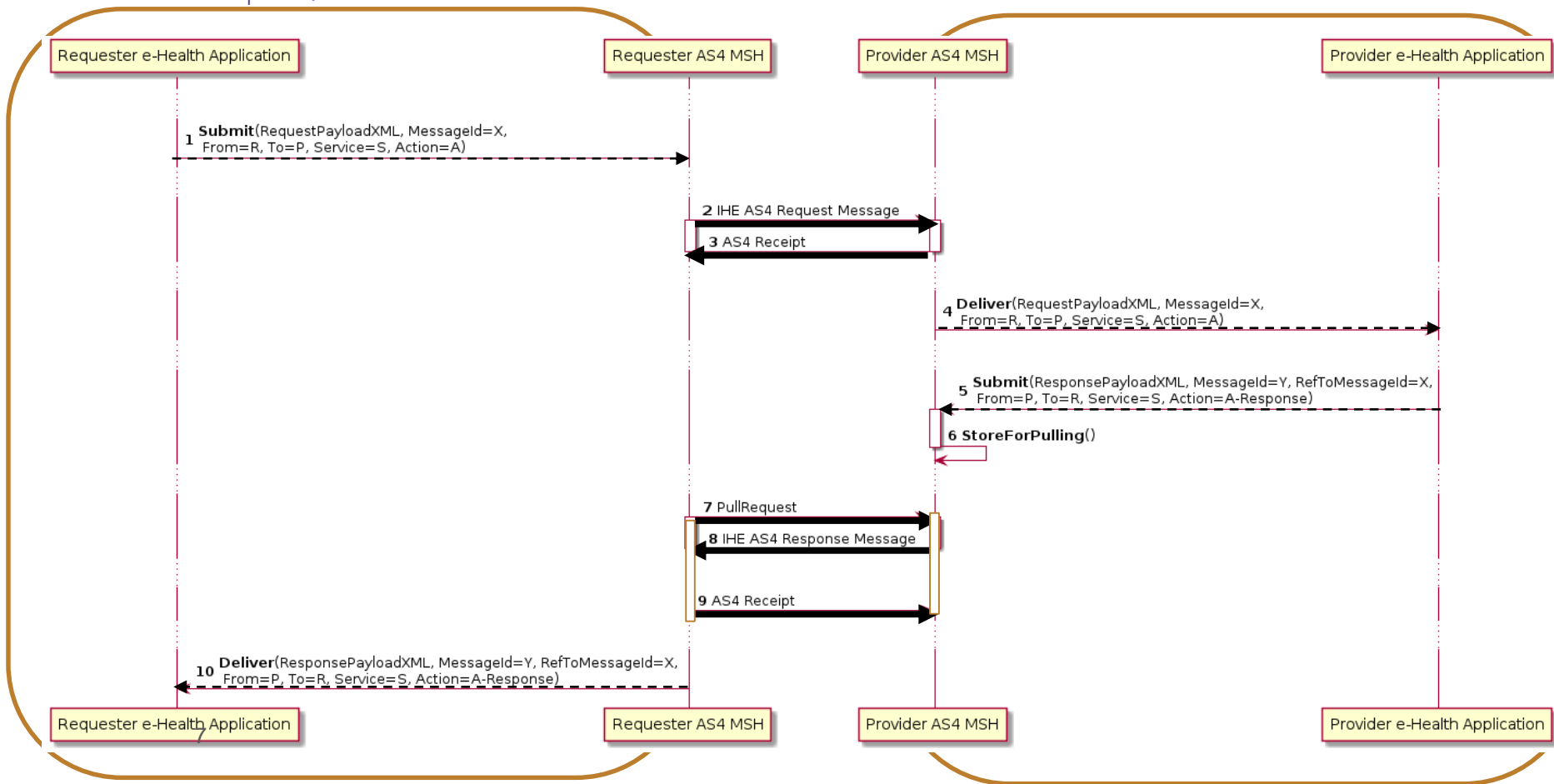
AS4 Message Exchange Patterns (MEP)

- Two Way MEP, reflect the fact that IHE transactions typically follow Request / Response pattern
 - Potential future use of One Way MEP (IHE DSUB profile)
- In a MEP, each leg may be configured to use Push or Pull binding
 - Current IHE stack uses Push for request, response on backchannel
 - Pull feature helps address network connectivity constraints (firewalls policies) on incoming connections; known problem for projects willing to use current (WS-Addressing) asynchronous exchanges based on the IHE technical framework
- For (evolutions of) implementations of current IHE stack, Push-and-Push MEP likely initial target
 - Push-and-Pull, Pull-and-Push new alternative opportunities

Two Way Push-and-Push Message Exchange Pattern



Alternative Two Way Push-and-Pull Message Exchange Pattern

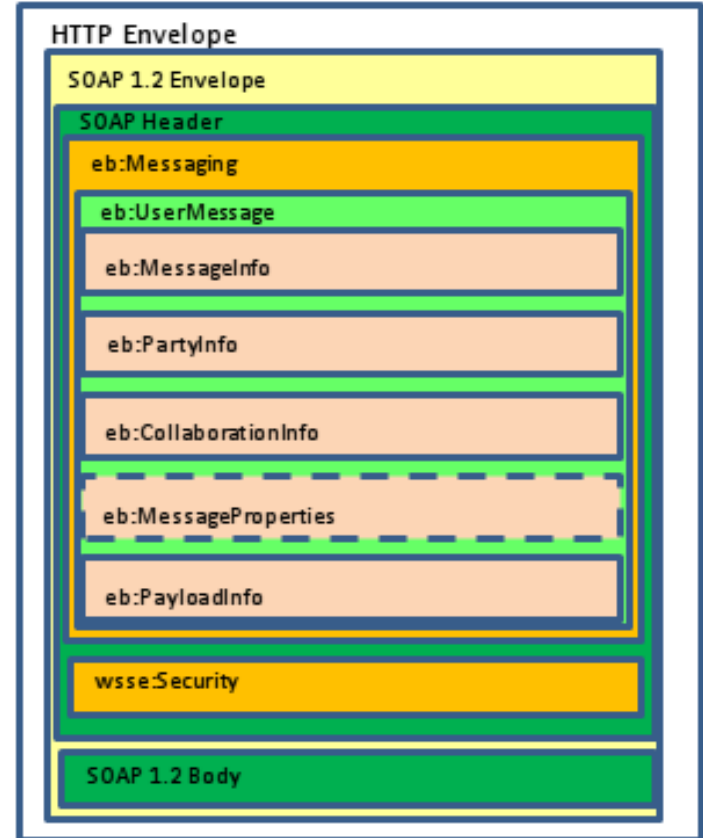


Packaging

- Request and Response XML content is carried in SOAP 1.2 Body
 - As in current IHE WS-Addressing based synchronous stack
 - Avoids many changes in IHE Technical Framework documentation and therefore preferred to stimulate adoption
- Use or non-use of MIME is profiled per transaction:
 - SOAP-with-Attachment MIME envelope for transactions involving binary “documents”
 - Simple SOAP 1.2 envelopes (no packaging of SOAP envelope in a root MIME part) for transactions involving only XML request or response
- Transactions involving documents in attachments involve cross-references from XML request/response to documents. It is done two ways:
 - Use of Part properties in AS4 PayloadInfo is the clean, native AS4 approach
 - Use of “xop:Include” and “xds:Document” is redundantly required to ease migration/adoption of existing IHE products.

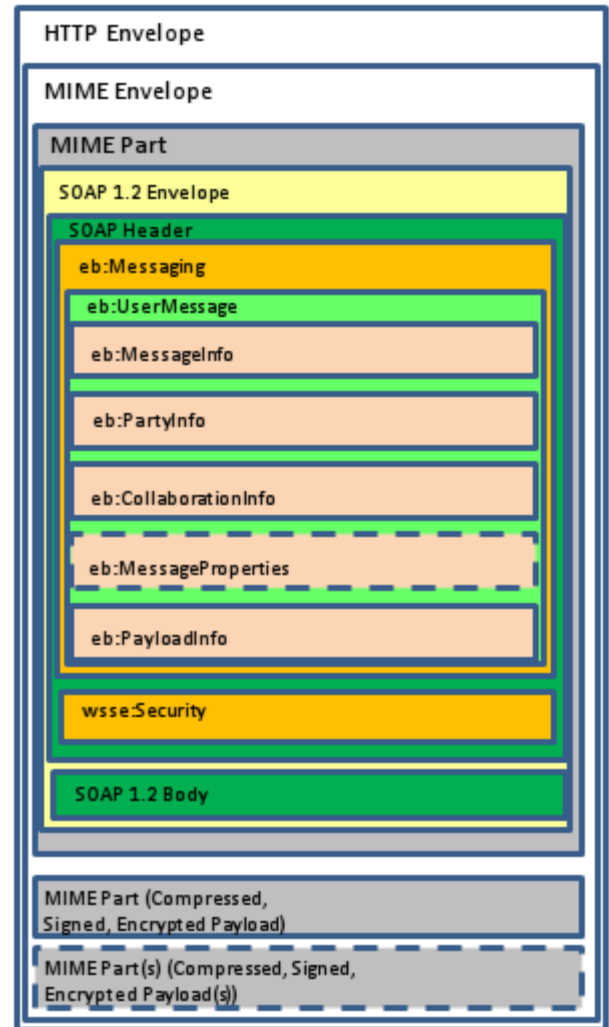
Simple SOAP 1.2 Packaging

- Used for transactions that do not involve (binary) “documents”, but only contain an XML request or request
- Simple SOAP 1.2 packaging, no use of Multipart/Related MIME Envelope
- XML request or response is contained in SOAP Body, as in current IHE stack



SOAP-with-Attachments Packaging

- Used for IHE transactions in which binary payloads (“documents”) are exchanged together with, but separate from, XML request or response
 - Cross Gateway Retrieve [ITI-39]
 - Provide and Register Document Set-b [ITI-41]
 - Retrieve Document Set [ITI-43]
 - Cross-Gateway Document Provide [ITI-80]
- SOAP envelope containing *eb:Messaging* header is in the SOAP root part of a Multipart/Related MIME envelope
- Binary documents are in natively compressed formats, no need for additional compression
- SWA replaces MTOM packaging in current IHE stack
- XML request or response is contained in SOAP Body, as in current IHE stack



Associating AS4 Payloads to Metadata

```

<eb:PayloadInfo>
  <eb:PartInfo>
    <!-- The first part is the XML XDS-b
         ProvideAndRegisterDocumentSetRequest document.
         Absence of an @href indicates the content is in
         the SOAP Body.
    -->
  </eb:PartInfo>
  <eb:PartInfo
    href="cid:0e3f6331-b5a8-4758-8cfd-c562d2eal1c86@requester.ro" >
    <!-- the first document in the package (PDF) -->
    <eb:PartProperties>
      <eb:Property name="id">Document01</eb:Property>
    </eb:PartProperties>
  </eb:PartInfo>
  <eb:PartInfo
    href="cid:9cf5c59a-068c-4c4d-a3ed-a24becee643f@requester.ro" >
    <!-- the second document in the package (JPEG) -->
    <eb:PartProperties>
      <eb:Property name="id">Document02</eb:Property>
    </eb:PartProperties>
  </eb:PartInfo>
</eb:PayloadInfo>

```

```

<env:Body>
  <xdsb:ProvideAndRegisterDocumentSetRequest
    xmlns:xdsb="urn:ihe:iti:xds-b:2007">
    <lcm:SubmitObjectsRequest>
      <rim:RegistryObjectList>
        <!-- details omitted -->
        <rim:ExtrinsicObject id="Document01">
          <!-- details omitted -->
        </rim:ExtrinsicObject>
        <rim:ExtrinsicObject id="Document02">
          <!-- details omitted -->
        </rim:ExtrinsicObject>
      </rim:RegistryObjectList>
    </lcm:SubmitObjectsRequest>
    <xdsb:Document
      id="Document01"><xop:Include
        href="cid:0e3f6331-b5a8-4758-8cfd-
        c562d2eal1c86@requester.ro"/>
    </xdsb:Document>
    <xdsb:Document
      id="Document02"><xop:Include
        href="cid:9cf5c59a-068c-4c4d-a3ed-
        a24becee643f@requester.ro"/>
    </xdsb:Document>
  </xdsb:ProvideAndRegisterDocumentSetRequest>
</env:Body>

```

Transaction Specific Profiling (1)

- Per transaction, for the request and the response, values are specified for:
 - Service
 - Action
 - From/Role
 - To/Role
- For Service and Request action, by convention, the same values are used as for `wsa:Action` in current stack
- The Response action is the same as Request action, with *Response* appended
- Resulted in a large number of changes to technical framework, but the changes are predictable as they all follow the same pattern

The requirements for the request and response messages with the AS4 Asynchronous Web Services stack are:

- The `<eb:Service>` SOAP element shall be set to the value: `urn:ihe:iti:2007:RetrieveDocumentSet`
- The `<eb:Action>` SOAP element shall contain the value: `urn:ihe:iti:2007:RetrieveDocumentSet` for the request and the value: `urn:ihe:iti:2007:RetrieveDocumentSetResponse` for the response message
- The `<eb:From/eb:Role>` element shall be set to the value: `urn:ihe:iti:2018:Requester` for the request message and to the value: `urn:ihe:iti:2018:Provider` for the response message.
- The `<eb:To/eb:Role>` element shall be set to the value : `urn:ihe:iti:2018:Provider` for the request message and to the value: `urn:ihe:iti:2018:Requester` for the response message
- The `<soap12:Body>` shall contain one `<ihe:RetrieveDocumentSetRequest>` element for the request message and the value: `<ihe:RetrieveDocumentSetResponse>` for the response message.

IHE Trial Implementation Process

- The Supplement: “Asynchronous AS4 Web Services Option” was developed by the IHE IT Infrastructure Committee between October 2017 and July 2018.
- It is now in *Trial implementation* and is publicly available from the www.IHE.net website: (https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_Suppl_AsyncAS4Option.pdf)
- Groupable with other profiles (e.g., XUA, DSG). Designed to not disrupt existing architectures
- The work item was used as an opportunity to re-document outdated text in the technical framework (Redocumented Appendix V in Volume 2.X of the ITI Technical Framework)
- The AS4 Supplement will be tested at the 2019 Connectathons in North America (January 2019) and Europe (April 2019).

Questions